Pelori

Designing a digital service
for maker projects through research

Master's thesis
Taro Morimoto

Media Lab Helsinki
School of Arts, Design and Architecture
Aalto University

2015

Aalto University
School of Arts, Design
and Architecture

Aalto University, P.O. BOX 11000,
00076 AALTO
www.aalto.fi

Master of Arts thesis abstract

Author Taro Morimoto

Title of thesis Pelori - Designing a digital service for maker projects through research

Department Department of Media

Degree programme MA in New Media

Year 2015 **Number of pages** 103 **Language** English

Abstract

The maker culture consists of people referred to as "makers", who are designing and building physical objects and intangible services through experimentation, innovation and knowledge sharing. This culture is thought to spur innovation and function as a catalyst to materialize the promise of the so-called new industrial revolution. Maker activities are centered on the projects they make. Makers use various digital services to show their work and exchange knowledge.

This thesis investigates what kind of digital services makers use when working with their projects. It describes a process of designing through research a new digital service that could complement their existing digital ecosystems.

The methodology used is participatory design and iterative software development. Expert interviews and co-design sessions with makers are used to create design directions for a digital service prototype for maker projects. The implemented prototype is then evaluated and re-designed to further develop a beta version.

The designed and implemented digital service is a production that is made in collaboration with Koert Jobse.

Keywords maker culture, digital services, mobile, human-computer interaction, software development, service design, user experience design

Contents

Acknowledgements

I want to express my deepest gratitude to my thesis partner Koert Jobse with whom I worked to create Pelori. Making the thesis and the production was a long project and I am thankful for all the numerous discussions and good times we had.

I am also thankful to my supervisor Kari-Hans Kommonen for all the great feedback and guidance in making the thesis.

I am grateful to Timo Nyberg for the advices and for motivating us to go forward. I also want to thank Timo and Social Manufacturing FiDiPro Project for providing us much needed funding to make it possible for us to travel to Maker Faire UK to evaluate the prototype with a large number of makers.

I want to thank Rasmus Vuori, Teemu Leinonen, and Cindy Kohtala for giving lots of feedback.

I want to thank Hacklab, Kaupunkiverstas, Aalto Fab Lab, the expert interviewees, and co-design session makers as this thesis and the related production could not have been possible without you.

And since I can only mention so many people here, I want to thank all the people who participated and helped me along the way.

Finally, I want to give a special thanks to my wife for supporting me in various ways during the process of making this thesis.

1 Introduction

This thesis investigates maker culture and the digital services used by its members and tries to find a gap or a weak point in the current digital ecosystem. As a production-based thesis it describes a methodology and a process of making a digital service for maker projects.

The production is done in collaboration with Koert Jobse, a master's student in Collaborative and Industrial Design at the Aalto University. Our individual theses look at the joint production from different points of view. This thesis looks at the production from a software development point of view. Unless explicitly mentioned, I have produced all content in this thesis. All content in the appendices is produced together with Koert except for appendix F that is written by Koert.

The source code of the digital service can be found at:
https://github.com/taromorimoto/pelori ("GitHub: Pelori" 2015).

1.1 Background and motivation

People have been making things from primitive to modern times. From primitive hunting tools to toys for children before toy stores weren't still around. People also fixed and modified tools when the requirements changed or when they found better or new ways of working. People have always responded to needs and challenges in their environments to make their lives easier and better. As Anderson points out, we are all makers as all of us make stuff whether it's drawing, cooking, gardening, or coding (Anderson 2012).

As the world has become more complex and more people inhabit the planet so has the amount of new ideas and ways of making things drastically increased. However, in the past, inventions have been local and hard to spread globally. Before the modern age the means to spread knowledge was limited to physical barriers. Knowledge spread from masters to apprentices as well as through books. Later the introduction of radio and television allowed for mass transmission of information from a central source to the masses. The biggest change happened at the end of last century when advances in digital and information technology enabled people to share knowledge much more efficiently and free of censorship and control. The key enabler has been the digital services that computers and the Internet have made possible.

Out of this new age of digital technology the maker culture was born and digital services enabled it to thrive to the point where it is today. Millions of people who were previously disconnected are now connected through the Internet allowing them to share projects, ask and answer questions, and discuss new ways of making. New ideas and innovations are being generated and shared like never before in history.

There are millions of great ideas and interesting work hidden in people's minds. These millions of locked away ideas and the digital services that have made the current wave of knowledge sharing and inventions possible inspire this thesis. This thesis investigates what kind of digital services could make this culture thrive even more. Specifically, what kind of elements digital services for maker projects would benefit from?

1.2 Personal motivation and production process

As the content of this thesis is in the realm of human-computer interaction (HCI) research, like Lindtner et al. (2014), I'm also personally motivated by the fact that the production in this thesis is not constrained to a lab but is out there in the real world used by members of the maker culture (Lindtner, Hertz, and Dourish 2014).

I've always been fascinated with ideas and how to find better ones. Around 10 years ago I had an idea of a digital service where people could post their ideas and people could comment and rate them. This would allow for new ideas to be tested and peer reviewed to help distinct good ideas from bad ones.

A year ago I started my work on this thesis. It has transformed a lot during the process of making it.

During the summer and early fall 2014 I designed and implemented a simple prototype to illustrate the idea. Idealizer, as it was called, was a web service that allowed users to post designs of anything and base them in practices and topics. It was also possible to compare the design side by side. Figure 1 shows the main view of Idealizer listing different kinds of lasagna designs.

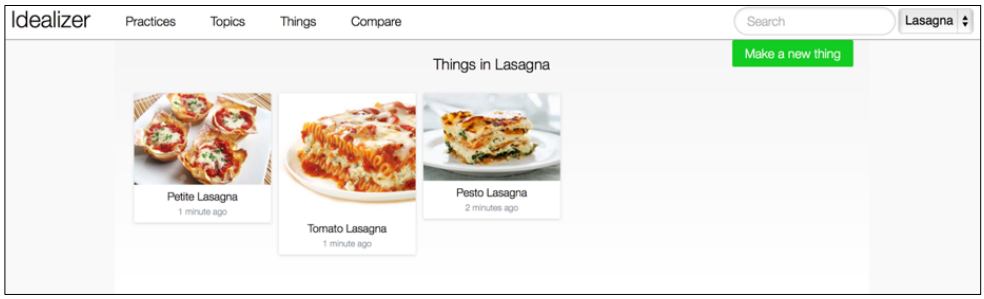


Figure 1: A working prototype of the Idealizer – "An interactive system for making better designs together".

As illustrated in Figure 2, the idea was to enable users to evolve designs over time with the help and contributions by other users. The goal was similar to services like GitHub but just on the textual and picture level without complex version control functionalities.

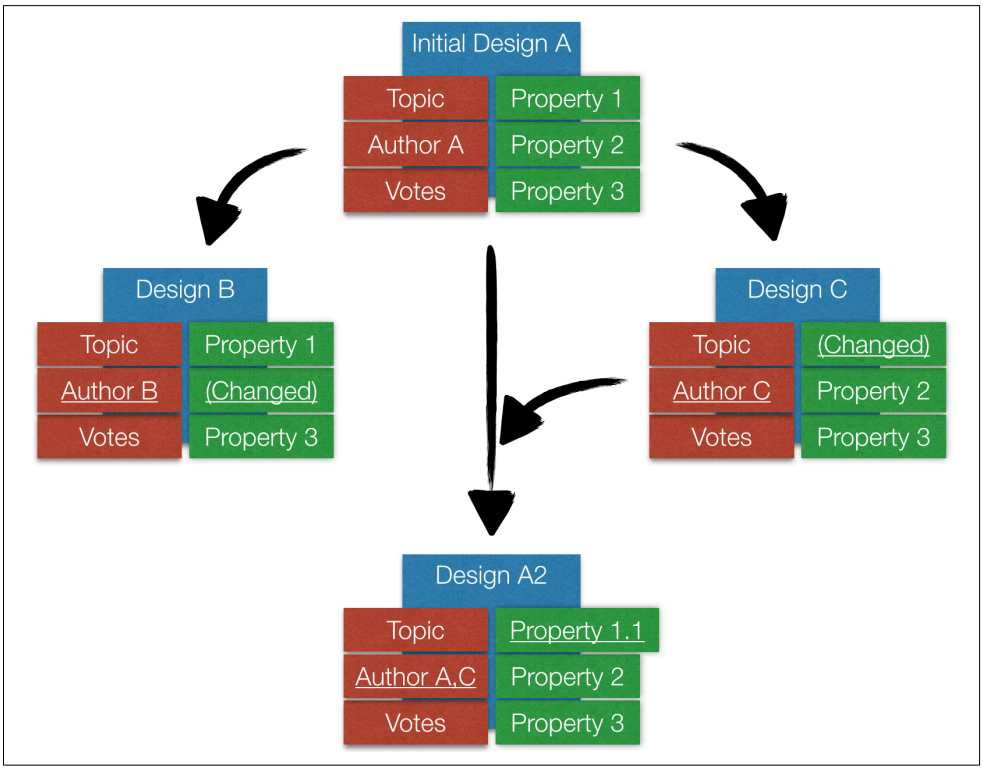


Figure 2: Versioning a design in Idealizer.

In the fall of 2014 I met Koert Jobse during a workshop called "Redesign of Society". Later that year we met by accident and found out that both of us were thinking about writing our master's theses around a similar topic. We ended up working together to create a joint thesis with a bigger outcome than just working alone. The combination of my software design background with Koert's visual design background made it possible for us to make a joint production in the form of a fully functional and polished digital service.

We then set out to come up with a service that we wanted to build. That included finding out for who we wanted to create it for and why. After multiple iterations of possible target groups and different ideas for a service we ended up designing for the maker community and their projects. This focus then allowed us to go deeper into what the service could be and specify what kinds of problems the target users had.

First step was to understand the stakeholders and the community. To do this, we visited maker spaces, conducted interviews, and organized co-design sessions. The interviews included 14 experts from researchers to professionals. We had 5 co-design sessions with each session consisting of one maker and us. This user research with members of the maker culture gave us insights into how they work, the tools and services they use, people they collaborate with, and the problems they encounter.

With the design directions we found we then set out to design and implement a prototype of a digital service that we ended up calling Pelori. As described in more detail in chapter 5 and Appendix C this prototype was intended to allow users to:

- Document easily

- Collaborate
- Share their projects
- Search and discover
- Enhance their digital ecosystem

The prototype is a smartphone application for iOS and Android phones that we then subsequently evaluated at the Maker Faire UK, the biggest Maker convention in Europe (Appendix D). At the faire, visitors were able to try out Pelori and at the same time we were able to talk to them. From this we got a lot of feedback and new ideas from the members of the maker culture.

After analyzing the results of the prototype evaluation we then re-designed Pelori and implemented a beta version of the application. A more detailed description of the redesign is shown in chapter 6 and Appendix E.

1.3 Research question

This thesis aims to find answers in the area of HCI to the following research question: **What kind of digital services could help makers improve collaboration in complex collaborative design projects?**

To seek answers to this question a digital service was designed and implemented through research and then studied whether or not it can perform as expected. This process was split into four major parts.

1. In the background research, literature was used to understand the maker culture better. Additionally benchmarking was used to map out the existing digital services makers use.
2. Expert interviews and co-design sessions with makers were used to find out if there are any gaps or weak points in the digital ecosystems the makers use for their projects.
3. Gathered knowledge was then used to design a prototype with elements that can improve collaboration in maker projects. A prototype of a digital service was then implemented and finally analyzed against competing services to see whether it can solve some of the problems better than existing services.
4. Based on the prototype evaluation, a beta version was designed and implemented. Finally the beta version was evaluated against a limited number of similar digital services.

The ultimate goal of the thesis and the production was to create a digital service that is useful for members of the maker culture and specifically for maker projects.

1.4 Thesis scope

This is a production-based thesis with a main focus on the production of a digital service and as such has a limited amount of literature review. The main focus is placed on software development process, software design, and software

production. More specifically, an iterative software development methodology is described in chapter 2. Thesis iterations are described in detail in chapters 3-6.

The background research chapter examines the maker culture from the perspective of innovation, openness and industry. Then digital services in the context of maker projects are investigated and a selection is benchmarked.

Although important, the different theories of human activities that take place in maker projects are left out of this thesis. These topics such as maker identities, maker spaces, knowledge conversion and transfer, and maker behavior are discussed in Koert's thesis (Jobse 2015).

Personas and use cases are not part of this thesis, however, an example persona developed by Koert can be found in appendix F. Other personas and use case related work are found in Koert's thesis.

Although crucial for any digital service that needs a large community, the question how applications spread among users is not discussed in this thesis.

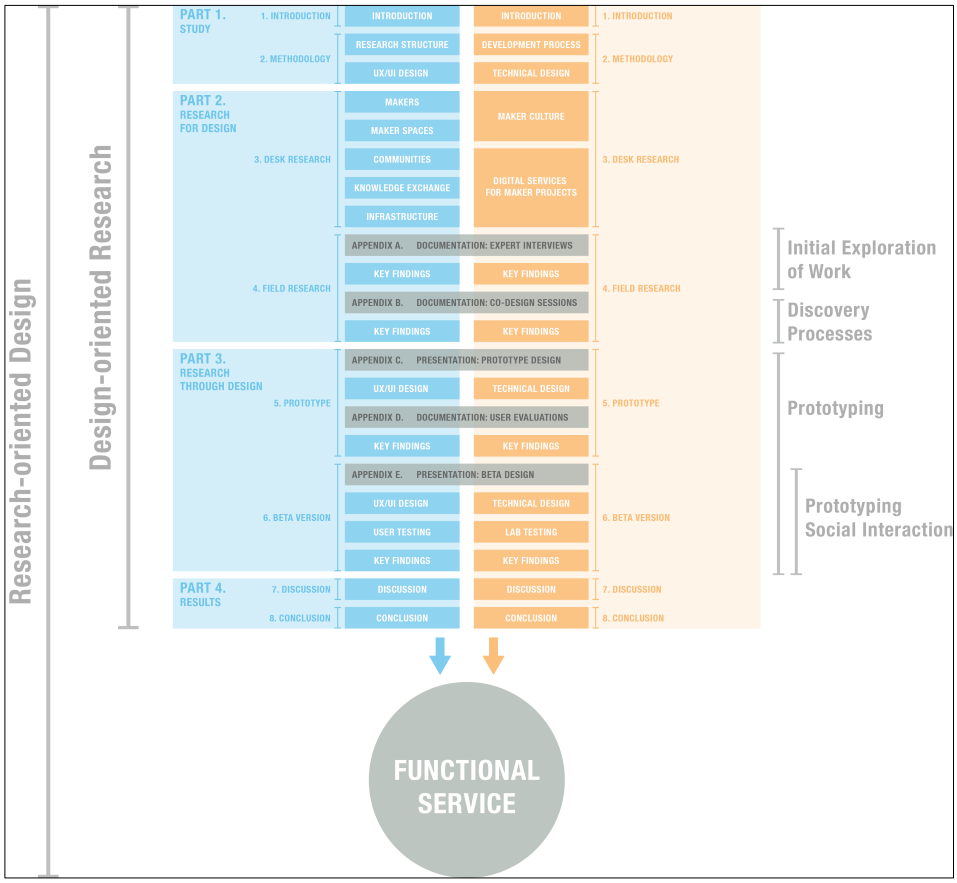


Figure 3: Research structure developed by Koert shows the division of work between this thesis, Koert's thesis, and the production.

The comparison between this and Koert's thesis is shown in Figure 3, which shows that the overall structures of the theses' are very similar but that the chapters can differ significantly. Some of the chapters in both theses have the same title but the content in those chapters is still produced independently. For example some of the

key findings chapters use the same results in the appendices but analyzes it from personal and individual thesis's point of view so as a result they can differ significantly. Unless explicitly mentioned, I have produced all the content found from the body of this thesis. Figure 3 is made by Koert and the research structure described in chapter 2.1 is developed by Koert.

Most of the work in the production has been done together with Koert Jobse. This includes interviews, co-design sessions, concept design, and functional parts of the UI/UX design. Koert is responsible for the visual UI/UX design as well as the logo and brand design. I am responsible for the technical design and for majority of the technical implementation. The exact code contributions per person can be found from GitHub at https://github.com/taromorimoto/pelori/graphs/contributors ("GitHub: Pelori" 2015). The documented results of the common work done in the project can be found in the appendices with the exception of appendix F that contains an example persona from Koert's thesis.

In summary, my and Koert's theses are independent works with a common production.

1.5 Thesis outline

This thesis has three distinct main parts, introduction and motivation in chapter 1, production in chapters 2-6, and discussions in chapters 7-8.

Chapter 1 introduces maker culture and lays out a personal motivation for the thesis and the production.

Chapter 2 describes the methodology and the development process used for the production.

Chapter 3 describes relevant topics and analyzes it through existing research. A number of relevant digital services are listed and a selection is benchmarked.

Chapter 4 analyzes the interviews and co-design sessions and creates design directions for a prototype. Out of this and the background research design directions are then presented on how to design a digital service for maker projects.

Chapter 5 reviews conceptual, visual and technical design directions created respectively by Koert and me. Based on the design directions a technical design is then created and described. After the prototype is ready, feedback is analyzed from two events where the prototype was evaluated. A re-design of the prototype is then described.

Chapter 6 describes the implementation and evaluation of a beta version that was created from the prototype re-design.

Chapter 7 reflects on the process of the entire project and describes the learning outcomes of the process. It also discusses the future directions for research in the topic as well as lays out a development road map for the created digital service.

Chapter 8 summarizes the research, analyzes findings from the production and answers the research question posed in chapter 1.

2 Methodology

The target is to design and construct a tool that enables makers to collaborate during their projects. This tool or a digital service can be a mobile app or a web service that assists makers in preserving and sharing knowledge.

In order to achieve this we have chosen methodologies that keep the users close to the development process so that it's possible to learn from and generate new ideas with different kinds of makers. More specifically, interviews and co-design session are conducted, and prototypes are tested with makers. With this a digital service is designed and implemented, and finally a beta version is released.

Since this is a thesis about a production, it can be said that the thesis is a design-oriented research and the production is research-oriented design (Fallman 2009). In this sense there are two processes with two methodologies running in parallel to each other. One is the research in the thesis and the other is the development process in the production. However, it should be mentioned that these two processes are very much entangled and it can be difficult to know where they are apart and where they are together. As Fallman argues, having a project that positions itself somewhere in between research-oriented design and design-oriented research is not optimal for most HCI projects since it might be too much to try to do both good design and good research.

Hence the ultimate goal of the thesis and the production is to create a fully functional digital service for maker projects that is useful and usable for makers. The aim is to end up with a finished enough digital service that can be sent out to makers for beta testing.

2.1 Research methods

Thesis research structure developed by Koert, shown in Figure 3, uses participatory design that continuously involves users in design, development, and evaluation efforts (Blomberg and Henderson 1990; Cardenas-Claros and Gruba 2010; Spinuzzi 2005). The model in Figure 3 shows how three participatory design research stages are applied in this thesis:

1. Initial exploration of work
    a. Expert interviews
2. Discovery process
    a. Co-design sessions
3. Prototyping
    a. Prototype
    b. Beta version

These participatory design methods work well with the software development methods chosen by me. Those methods are described in the next chapter.

2.2 Software development methods

The production team consists of two people that work very closely together in the same location practically every day. This means that development process and project management is kept as light as possible. The methodology chosen for the production development process is a loose form of agile and iterative development (Larman 2004; “Manifesto for Agile Software Development” 2015).

More specifically no strict formalities such as Scrum daily stand-up meetings or sprints are used. The project schedule is an overview of the project for each month written down in a collaborative text document in Google Docs (“Google Docs” 2015). Task and issue management for the prototype and the beta version is done through Trello boards (“Trello” 2015). Source code and version control management service GitHub is used to handle the practical issues that occur when developing the service with two people (“GitHub” 2015).

The production consists of four major iterations. At the end of each major iterative cycle the developed design and findings are evaluated and used to gradually improve the overall design. Below is a list of are the major cycles.

1. Background research including benchmarking existing digital services.
    a. To develop a deeper understanding of the target field and competing digital services.
2. Expert interviews and co-design sessions with makers.
    a. To gain deeper knowledge on relevant areas of interest.
    b. To gain better understanding of the users, their problems, and current solutions.
3. Prototype design, implementation, and evaluation.
    a. Develop a working prototype and get feedback from target users in a controlled setting.
4. Beta version design, implementation, and evaluation.
    a. Develop a beta version that can be sent out for beta testing to target users in a real world context.

3 Background research

The aim of this chapter is to broaden the understanding of the culture that the digital service is going to be designed for.

3.1 Maker culture

"We are all Makers. We are born Makers and many of us retain that love in our hobbies and passions. It's not just about workshops, garages, and man caves. If you love to cook, you're a kitchen Maker and your stove is your workbench. If you love to plant, you're a garden Maker. Knitting and sewing, scrapbooking, beading, and cross-stitching—all Making."

– Chris Anderson (Anderson 2012)

People have always been creating new objects and fixing and modifying old ones. In the modern times this culture has generally been called the "do it yourself" (DIY) culture. The term "maker" appeared after Dale Dougherty created the MAKE magazine and Maker Faire events ("Make" 2015; "Maker Faire" 2015). As Mota explains it, a subset of DIY community who were involved in digital technology and/or hardware and technologically enhanced arts and crafts became knows as the maker community (Mota 2011). These makers were empowered by the early promise of digital fabrication and digital services that now connected millions of makers together (Mota 2011; Anderson 2012; "A Movement in the Making" 2014).

Now that the term maker has been out there for some time, it appears as the definitions of DIY culture and maker culture have started to closely resemble each other. Koert brings together different definitions of DIY and maker culture and argues that these two terms are used to describe the same thing.

Maker activities can be thought of as the activities that happen in the maker culture. A subset of this, called making activities, can be thought of as an activity that involves creation and/or repairing objects and/or tools. At the core of making activities is knowledge of how to work with materials and who to use different tools. Sharing this knowledge has been done through many different media from person-to-person communication to magazines and books, and now also through digital services on the Internet (Anderson 2012).

Maker spaces where maker activities happen can be categorized into academic, commercial, community driven, and governmental supported, as well as hybrid maker spaces (Jobse 2015). In addition to these types of maker spaces, spaces such as garages or toolsheds could be thought of as personal maker spaces. Most of these spaces are not just places for laser cutting or 3d printing, but also places to experiment on new ways on how communities work and collaborate together (Lindtner, Hertz, and Dourish 2014).

Despite the open nature and inclusive goals behind the maker culture and the fact that there are now maker spaces all over the world, the reality is that the makers are

a relatively homogenous group consisting mainly of middle or upper class men (Ames et al. 2014).

3.1.1 The New Industrial Revolution

Anderson talks about the new industrial revolution where in todays world "bits" on personal computers can be easily turned into "atoms" in on-demand "factories" in a small or a large scale (Anderson 2012). These bits represent the designs of physical objects and the atoms manufactured physical objects where factories are either digital desktop fabrication tools or remote-manufacturing services that produce and ship the objects based on digital design files. According to Anderson the technologies that have made this possible have been developing for the past two decades and the next decade will be about making the revolution a reality.

Makers, who are the pioneers of this new revolution, can be categorized into three separate groups: zero to maker, maker to maker, and maker to market ("A Movement in the Making" 2014). People in the maker to market category are the ones who are commercializing the things they are making. They are a group of people who are taking their innovations and turning them into sellable products, extending their maker activities beyond tinkering in garages into small hardware startups as well as graduating into bigger companies (Lindtner, Hertz, and Dourish 2014; Ames et al. 2014).

According to Lindtner et al., maker spaces in China, the so called "innovation houses", are government funded and show an effort from the Chinese government to transition from manufacturing to innovating and designing products – in other words from "Made in China" to "Created in China" (Lindtner, Hertz, and Dourish 2014).

3.1.2 Innovation in the maker culture

"So collective invention is defined here to be ***a process in which improvements or experimental findings about a production process or tool are regularly shared****. Put this way, collective invention seems to be an important and regular feature of the historical process by which societies adapt to radically new technologies. It is part of a larger picture in which the new technology turns into new products and the producers, consumers, and markets are jointly developed."*

– Peter B. Meyer (Meyer 2003)

Technological inventions are often kept secret or protected with patents whereas scientific inventions are openly published. The motivation behind these technological inventions is usually driven by the need to make a profit. However, when a radically new technology arrives, a period of collective invention occurs where knowledge is shared more freely and over time the new technology is adapted to useful purposes becoming more useful to the society (Meyer 2003). Recently, as a foundation for the maker culture, it can be said that periods of collective inventions happened with microcomputers from 1975 to 1990s and with open source software from 1960s to 1990s.

Meyer formulates a model of collective invention where an opportunity to make use of transformative inventions creates opportunities for a period of innovation

where hobbyists and firms generate a flow of micro inventions (Meyer 2003). A period of collective innovation is a time of sharing findings and inventions and is marked with openness. Examples of Meyer's collective invention are shown in Table 1.

	Steam engine case	**Cleveland district iron blast furnaces**	**U.S. mass production of steel case**	**Microcomputer club, Homebrew example**	**Open source case Linux example**
Instigating or enabling events	Watt's patent expiration 1800		Patent pool agreement in U.S., 1867	Microprocessors available, 1971	Internet (circa 1970), AT&T breakup, (1984)
Common institutions or publications	Lean's Engine Reporter, 1811- 1904	Books and consultants, 1850s-1870s	Prof. journals like TAIME and Bessemer Assoc. publications, starting 1871	Homebrew computer club meetings and newsletter, 1975-1980s	Internet bulletin boards and the Linux source code itself, starting 1991
Price of entry or restrictions on entry	Not available.	Informal inclusion	$5000 till 1877, then $80,000. (Temin)	Zero	Access to Internet bulletin boards
Tacit knowledge, skills, and prerequisite tools	Steam engine engineering or operation		Investment capital and background in furnaces or rolling mills	Practical electronics knowledge	Unix development software (compiler, linker)
Readers	Mine managers in Cornwall region of southwest England	Iron makers in Cleveland district of northeast England	Bessemer patent licensees in U.S.	Silicon Valley computer hobbyists	Unix programmers, connected by the Internet
# of contributors	Approximately two dozen		Dozens	Many dozens	Thousands (Pavlicek, p. 63)
# of readers			Around 1000 AIME members	Several hundred	Thousands
Editor or moderator	Joel Lean, then his sons	Isaac Lowthian Bell, and others	AIME, other professional associations, and Holley	Lee Felsenstein, Gordon French	Linus Torvalds

Table 1: Episodes of collective invention (Meyer 2003).

When moving from the 20th to the 21st century, Baldwin and Hippel explain how innovation is expanding from a predominantly producer driven company-based model into single user and open collaborative innovation (Baldwin and Von Hippel 2010). More specifically, innovation in the 20th century was mostly done in centralized R&D and product development groups in the hands of companies that produced goods for consumers. In the 21st century innovation dispersed to a wider sector of the economy with the introduction of ever cheaper computing power, simplifying process technologies, modular design methods, and cheaper and instant communication technologies.

It can be summarized that what is currently happening in the maker culture is based on a series of technological innovations that have enabled knowledge sharing and collaboration through the Internet thus sparking this episode of collective and open collaborative invention in the maker culture.

3.1.3 Openness in the maker culture

According to a study by Kuznetsov et al. information exchange is a core value for DIY communities (Kuznetsov and Paulos 2010). For information exchange to happen a degree of openness is required.

Open source has been a transformational phenomenon for software development and one of the most successful stories of openness. Similarly, what open source is for software development, open design is for the maker culture. Open design, based in open source software and open hardware are enabling makers to share knowledge and increase innovation (Raasch, Herstatt, and Balka 2009).

Maker projects usually contain a fair amount of tacit knowledge, a certain kind of implicit knowledge involved in practices, that is difficult to formalize into explicit knowledge and transfer to another person (Polanyi 1966; Tsoukas 2005). Tacit knowledge is not only contained in one person's mind but can also collectively reside in the minds' of a community of people where it can be a source of innovation (Leonard and Sensiper 1998).

Majority of open source projects are digital and contain mostly explicit knowledge. In contrast open hardware projects usually contain a lot of tacit knowledge because the domain of hardware is physical. As an example it is very difficult to explicitly explain what kind of force and motion is required for soldering. Thus the benefits of openness in open source in software development are not always transferrable to open hardware.

Open Hardware and Open Design have many benefits, however, when it comes to mass manufacturing it can become difficult to hold on to the ideals of openness (Lindtner, Hertz, and Dourish 2014). An influential computer hobbyist group in Silicon Valley, The Homebrew Computer Club, is a good example of this where many of the members who started companies eventually left the club since they became too busy, gained enough prestige, or it was too uncomfortable for them to hold secrets in a culture where sharing and openness was an essential principle (Meyer 2003). More recently in 2012 one of the most successful open hardware startups, MakerBot, went closed-source with some of their new 3D printers and changed the terms of their successful 3D model sharing service (Lindtner, Hertz, and Dourish 2014).

3.2 Digital services for maker projects

Easy access and affordability of tools has been one of most important factors in a renewed interest in DIY cultures and practices, however, equally important has been the emergence of new sharing mechanisms in the form of Internet connected digital services such as social networks, forums, blogs, questions/answer services, and video sharing services (Kuznetsov and Paulos 2010).

A typical maker activity happens in a project, either alone or in collaboration with other makers. These maker activities like any other practices have an existing digital and design ecosystem already in place and introducing a new tool to makers' design toolkit should avoid disrupting existing designs and practices, and instead find ways to support and strengthen it (Kommonen 2013).

Maker projects vary significantly and depending on how strictly one wants to apply the definition, a purely digital project could also be called a maker project. However, for simplicity and to narrow the scope, maker projects in this chapter will primarily focus on projects that have some kind of physical aspects about them.

Makers use various digital services during and after working on a project. With a lack of research into which specific services they are currently using a hand picked selection is listed in Table 2.

3.2.1 Motivation and online communities

One of the key aspects of maker culture is the idea of empowerment which capacity is increased through lower barrier of entry and more accessible tools, that enable makers to make things on their own instead of relying on buying things (Grimme, Bardzell, and Bardzell 2014).

In addition to making things by themselves, makers also have a sense of contributing to the community and humanity as a whole, to give back and help others (Grimme, Bardzell, and Bardzell 2014; Kuznetsov and Paulos 2010).

A study of six DIY online communities describes the reasons why makers use digital services (Kuznetsov and Paulos 2010).

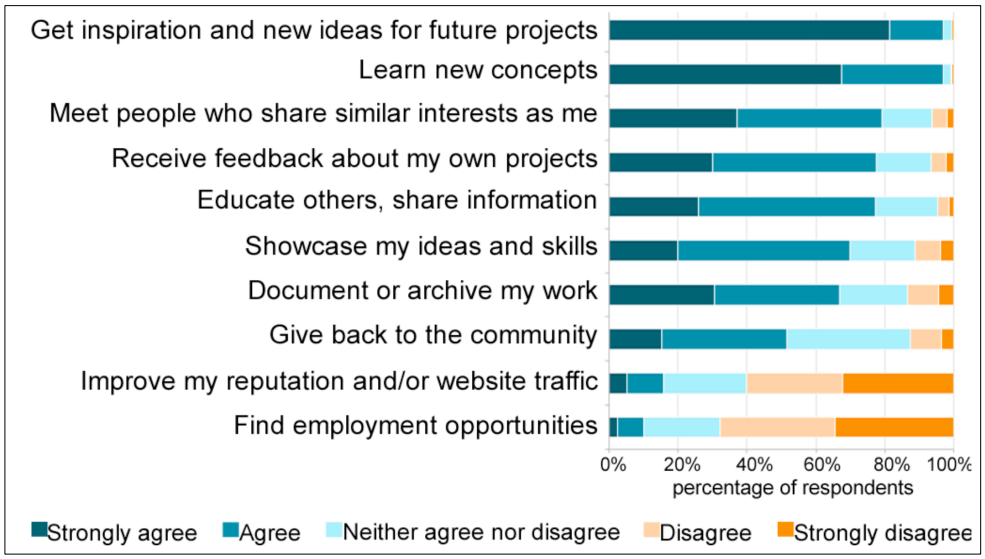


Figure 4: Motivations for contributing to DIY communities (Kuznetsov and Paulos 2010).

One of the things that make tools more accessible and lowers the barrier of entry is getting help from people who already know how to use tools and have worked on similar projects that beginner-makers want to make. In order to benefit from this knowledge, these advanced makers should share it. Figure 4 shows a survey into online DIY communities by Kuznetsov and Paulos that suggests that knowledge sharing (new ideas, learning, feedback, educating, showcasing work) is one of the biggest areas of motivation for contributing to DIY communities.

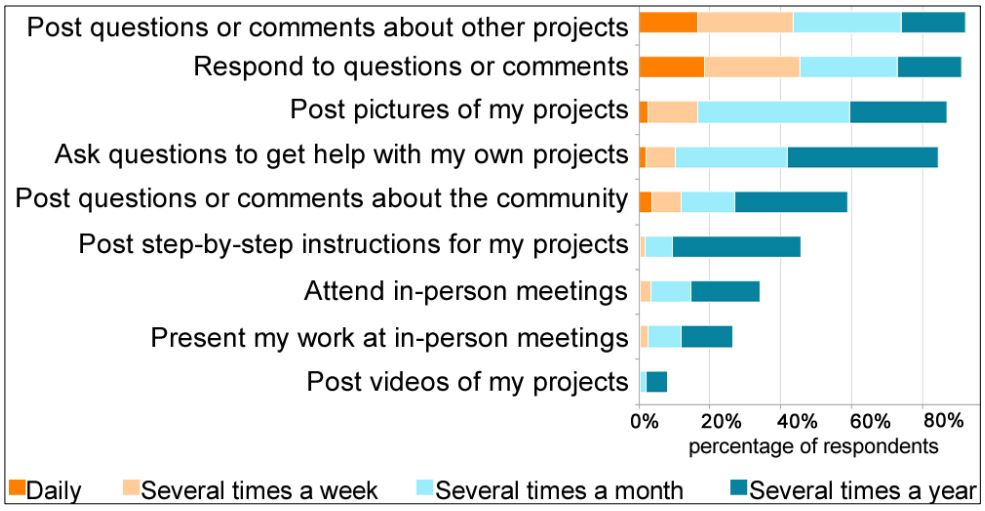


Figure 5: Contributions to DIY communities by frequency (Kuznetsov and Paulos 2010).

Makers contribute by participating in multiple online communities where they share their projects, and discuss with each other. As shown in Figure 5 the most common way of contributing in a DIY online community is by posting/responding to questions/comments about projects, posting pictures of ones own projects, and asking questions to get help on their projects.

3.2.2 Key elements of digital services for maker projects

This thesis focuses on 6 different key elements of digital services for make projects: taking notes, writing documents, capturing media, collaboration, ideation, and openness. The selection is based on the different aspects and activities involved in maker projects. Specifically elements that are part of activities that motivate makers and advance innovation are considered. It should be noted that this grouping is not a scientific formulation but based on experience working with the maker culture and findings from earlier chapters in this thesis. Some obvious key elements related to projects in general such as task and project management are intentionally left out.

Taking notes: Recording ideas and information in short textual format.

Writing documents: Writing extensively resulting in a polished text.

Capturing media: Documenting an object or a process with pictures, video, or audio.

Collaboration: Sharing information or working together on something with one or more people.

Ideation: Generating and developing ideas alone or collectively with a community.

Openness: One of the main elements in maker culture.

3.2.3 Benchmarking

Distributed and locally managed digital services have many obvious aspects that fit the maker culture's ideology such as open source, independence, and adaptation to local needs. One attempt to develop such a solution is a recent creation of FabMoment, a project documentation repository for individual Fab Labs, and a common meta language called FabML (a proposal for a common open hardware project description language), which hasn't been successful so far (Troxler and Zijp 2013; Määttä and Troxler 2011).

The digital services discussed and benchmarked in this thesis are mostly centrally managed opposed to distributed and locally managed. The apparent reason for this is that locally managed services have not yet succeeded in providing the kind of long term reliability and good enough service that the centrally managed and commercial digital services have.

Table 2 lists key elements of selected digital services for maker projects and selected digital services in the market that are used by the makers. Some of the services have multiple key elements but for clarity most of them have been assigned into only one best fitting group.

Key element	Examples of popular digital services
Taking notes	Evernote, Notes for iOS and OS X, OneNote, Google Keep, EtherPad
Writing documents	Google Docs, Microsoft Word, OS X Pages, Instructables, Medium
Capturing media	Native camera and photo gallery applications, Instagram, Storehouse, Evernote, YouTube
Collaboration	Google Docs, GitHub, Facebook Groups, Slack, Trello, Internet forums, IRC, EtherPad, Thingiverse, Knowable
Ideation	Pinterest, Quirky
Openness	Quirky, Pinterest, Instructables, GitHub, Internet forums, Knowable, IRC, Thingiverse

Table 2: Categories of digital services for maker projects including example services.

There are many well-suited digital services that are useful for maker projects and could be benchmarked but in order to limit the scope only 4 services were chosen.

Table 3 shows a summary of each selected digital service to be benchmarked.

Service	Tagline	Description	Platforms
Evernote	The workspace for your life's work	As one workspace that lives across your phone, tablet, and computer, Evernote is the place you write free from distraction, collect information, find what you need, and present your ideas to the world.	Mobile, desktop, browser
Facebook Groups	Share what you care about with the people who care about it most	See all of your Facebook Groups in one place. Discuss, plan and collaborate easily and without distractions. Follow your groups here or on Facebook, whichever is easier for you.	Mobile, browser
GitHub	Build software better, together	GitHub is the best place to share code with friends, co-workers, classmates, and complete strangers. Over 10 million people use GitHub to build amazing things together.	Desktop, browser
Quirky	The invention platform	Quirky is a community company. Our mission is to make invention accessible. We believe everyone can be an inventor, and invention can happen anywhere.	Browser
Instructables	DIY How To Make Instructions	Instructables is a place that lets you explore, document, and share your creations.	Mobile, browser
Internet forums	Online discussion platform	Internet forums are places to discuss freely about anything; in the Internet there is a forum for everyone.	Browser
Knowable	Collaboration for hardware startups	Knowable is collaboration and project management tool for makers and product designers.	Browser

Table 3: Overview of digital services selected for benchmarking.

3.2.3.1 Evernote

Evernote is a multiplatform digital service for projects ("Evernote" 2015). It is essentially a service for taking notes and organizing them. Evernote has three

subscriptions levels: basic (free), plus, and premium. This benchmark looks at the basic version.

The main focus is on writing texts and lists as well as collecting relevant information for the project as web links, capturing handwritten notes, and taking photos. Evernote has an elaborate camera tool for processing captured documents to look as if they were scanned. Evernote can also a search function for digitized text form captures photos. For text editing there are basic editing tools from font formatting to creating lists and tables.

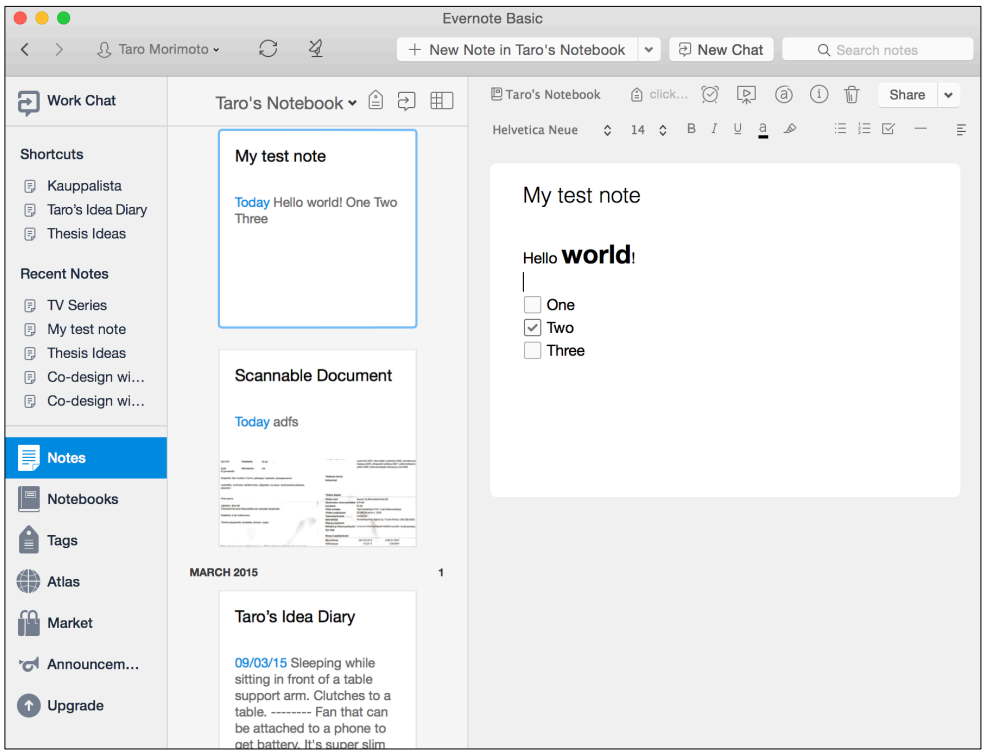


Figure 6: Desktop version of Evernote on OS X.

Evernote is truly multiplatform in the sense that in addition to mobile applications there is also a desktop and a browser version available. All of these versions retain practically all of the features with the exception of missing pictures capturing in the browser version. Notes are synced across all devices. Figure 6 shows the main view where in the center is a list of notes from a specific notebook and writing notes happen on the right.

The multiplatform nature of the application allows the user to work anywhere the user happens to be at that moment and whatever kind of notes she or he is working on. Quick notes can be easily made through the mobile application, and for longer writes or things that require more time the desktop version is an ergonomic and pleasant way to work.

Evernote has a lot of features and is very flexible to work with, however, this comes with a downside of cluttering the UI with many different buttons and making simple tasks seem more complex than they actually are (Figure 6). This allows experienced users to use the app very efficiently, however, a novice might find it a bit overwhelming.

Collaboration features in Evernote are fairly minimal. Although multiple people can have edit access to a note, in practice editing is restricted to one person at a time. It is also possible to select one or more Evernote users to send instant messages in a chat window.

The focus of Evernote seems to be for individual use. Specific notes or notebooks can be shared to specific people or be published as a public link. There are no community mechanisms in Evernote to see other users' public notes, meaning that searching and discovering other peoples interesting projects is not possible.

3.2.3.2 Facebook Groups

Facebook Groups is a subset of Facebook social network ("Facebook Groups" 2015). It is a collaboration tool for specific groups of Facebook users. In the service a group can be either private or public so that various kinds of groups can use the service. The main feature is the discussion feed with posts from members (Figure 7).

Figure 7: The browser version of Facebook Groups on OS X.

It should be noted also Facebook Messenger service has a section dedicated to group messaging, however, these messaging groups are separate from the actual Facebook Groups. These messaging groups consist of selected individuals that need to be hand picked every time a messaging group is created. In other words, groups in Facebook Messenger and Facebook Groups are not truly connected.

The discussion feed in Facebook Groups is almost identical to the Facebook News Feed with the distinction that it is specific to that particular group. In practice the discussion feed in Facebook Groups is very similar to the traditional Internet forums that contain posts and each post contains comments. Each post and comment consists of text, pictures, videos, and/or links to websites. Users can choose to get notifications when a Facebook friend or anyone creates a new post.

What makes Facebook Groups different from other Internet forums is that UI design is much more loose and that majority of Facebook users use their real names where as in the forums users are typically using avatars or anonymous usernames.

What makes Facebook Groups so powerful is the familiarity of the service to the public. In 2015 Facebook had 1.44 billion monthly active users ("Statista" 2015). Having such a huge user base, sharing mechanisms also work much better especially when connected to Facebook's News Feed.

Facebook Groups is a service that is made to cater for a wide range of groups from all walks of life and the typical context is social, ideological, or a specific field of interest. In the context of maker projects that kind of generalization doesn't make Facebook Groups an ideal place for maker activities. More specifically, in the case of one maker project per one Facebook group, it would be difficult to connect maker project groups together and therefore discovery would be minimal. Also there is no good way of searching all the maker projects in Facebook Groups.

3.2.3.3 GitHub

GitHub, established in 2008, is one of the most widely used open source software repositories with 10 million users collaborating over 24 million repositories ("Press · GitHub" 2015). GitHub uses Git, an open source distributed version control system, to host public and private repositories that allow one or multiple users to collaborate on projects ("Git" 2015). In addition to Git version control features GitHub also has issue tracking, documentation via Markdown-like README files, wikis and simple websites, and various usage data visualizations ("Markdown" 2015; "README" 2015).

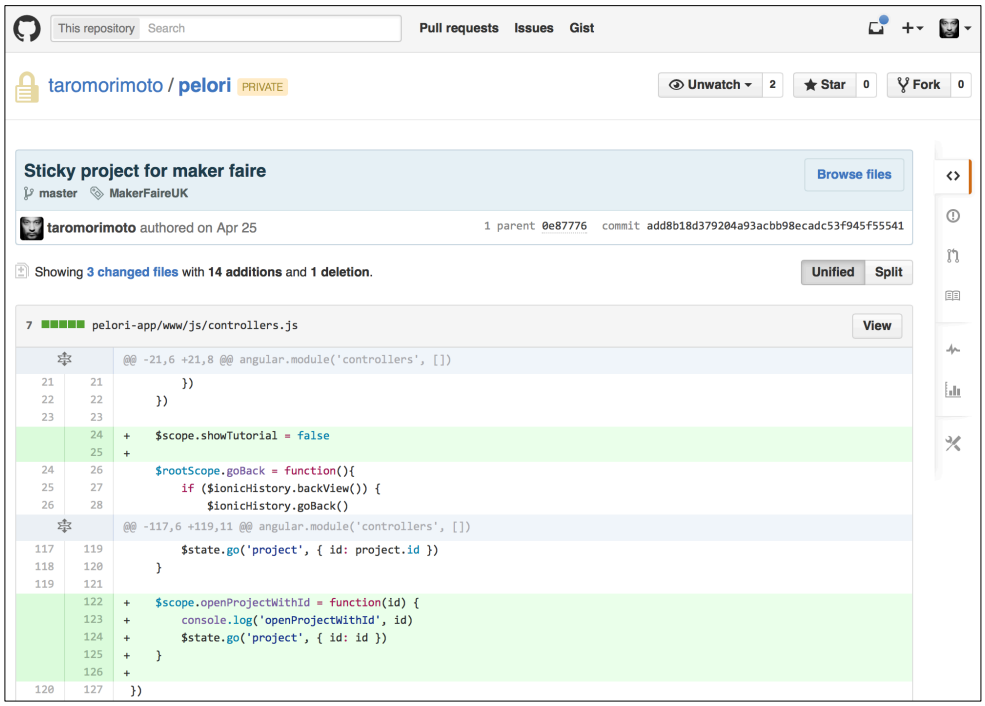


Figure 8: Showing changes in a commit in GitHub.

GitHub is especially good for distributed development where anyone can contribute to a project via Git pull requests, or a group of packaged changes, that can be discussed, modified, and eventually rejected or approved by project owners ("Git" 2015). An example of a commit (a change package) that can be made into a pull request is shown in Figure 8. The open source ideology behind Git works also in GitHub to allow projects to be forked to create new versions of them.

A user can also follow other users, and watch and star repositories to keep track of the ones he or she finds important. This data is then combined into a feed where all relevant activity from these users and repositories is shown. The feed only shows one line textual update of what happened and as such only serves to inform that something happened and not specifically what it was.

Although GitHub is a great tool for managing and developing software projects, it can also be used for various other kinds of projects. However, since Git is designed to handle textual data and can resolve per character differences in text, for binary data such as pictures it can only detect if the file has been changed and cannot merge changes in the same file. This makes GitHub slightly less useful for visual designers and there are now better suiting services for them such as Pixelapse ("Pixelapse" 2015).

Features on desktop and on browser are divided so that most of the Git version control features are in the Git desktop client and the rest of the features are in the browser. For non-programmer users, Git clients are notoriously difficult to use and as such can provide to be a deterrent for many maker projects.

To put it simply, GitHub is an extremely powerful tool for developing collaborative projects but can be intimidating and difficult to use and requires a long-term use for mastery. Addition to this it lacks a mobile application and as

such is not viable for multiple kinds of maker projects where a computer is not always available when working on a project.

3.2.3.4 Quirky

Being an invention platform for people to collectively invent products, Quirky positions itself squarely in the maker-to-market category ("Quirky" 2015). Quirky aims to make it as easy as possible for inventors to develop and make money out of their ideas. The platform is also not only for individual inventors but also for people who want to contribute to inventions by offering new ideas to take it forward or expert skills such as 3D modeling to improve the presentation of the invention. As shown in Figure 9, inventors can share influence points of the invention to get others to collaborate on the invention. Later on if the invention is produced and sold, people with influence points will earn a share of the revenue.

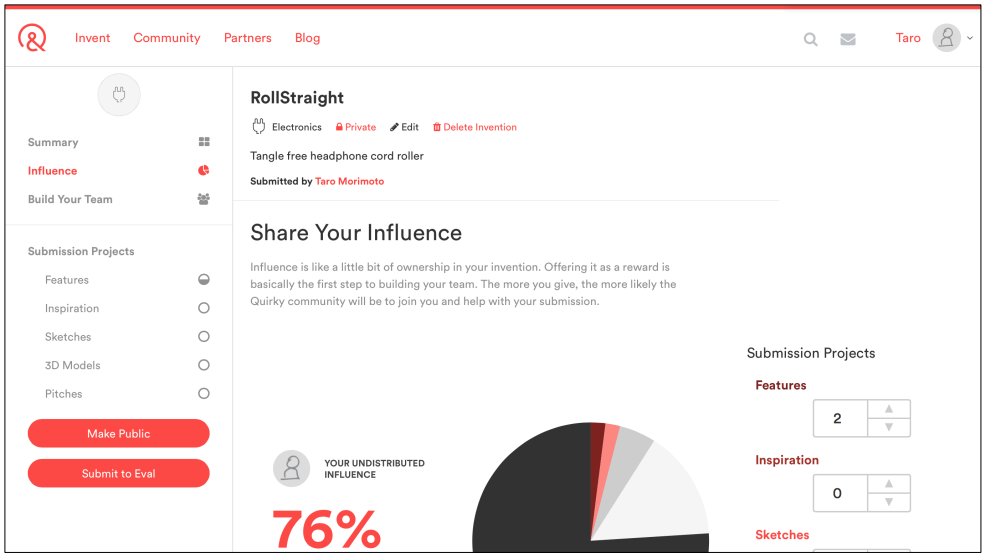


Figure 9: Sharing influence to get help from the community to improve your invention.

Addition to a collaborative invention, Quirky can also be used as a tool or a framework to develop an invention in private. Quirky help inventions have more structure and cover relevant aspects when shown to investors. Private inventions can also be shared to specific people to get feedback and add collaborators.

Quirky can only be used thought the browser so it isn't really meant to be used while working on a project away from the computer but rather sitting down on a desk and concentrating on the service itself.

Although Quirky seems to be very well thought out to incubate product inventions, the fact that it's purpose is to commercialize ideas limits the service's usefulness for maker projects.

3.2.3.5 Instructables

Instructables is one of the best maker project repositories and is specially designed and built for that purpose ("Instructables" 2015). Users can post projects to the

service that other users can then view, favorite, comment, and mark as "I Made It!" (Figure 10). Each project contains a description and a number of steps to complete the project. Content types include text, images, links to videos, and attached files. A project has tags and can be added to a collection or a group. Each collection has to be categorized on two levels with a category and a channel. Each group has Instructables, members, and topics (or discussions). It's also possible to follow users, groups, and collections. The service also has a wide range of different kinds of contests that you can follow or enter.

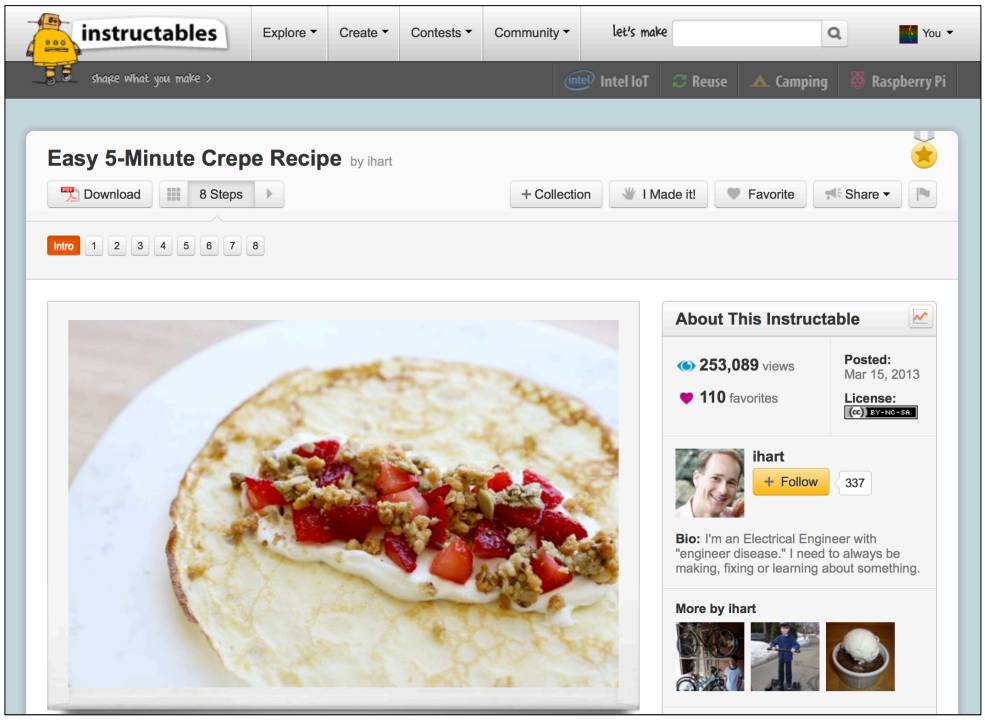


Figure 10: Project view in Instructables browser version.

Discussion in Instructables happens in various places such as forums, answers sections, group topics, and comments sections at the end of project views. Discussion can be branched by replying to an individual comment. The forums have a wide variety of predefined categories of discussions. Topics under each discussion are user created. Topics from groups also feed into the forum discussions. In the answers section each question can be followed.

Instructables has a lot of features, however, it is also something that makes the service unnecessarily complex. This complexity also makes designing the UI a difficult task, so elements like the feed of users and things a user is following is difficult to find from the browser version. Also many things are not very well though out such as the feed having only a big picture, a title, and the author's name, where time and a short description is omitted. This makes the feed appear more like a gallery rather than a timeline of activity from the sources a user is following.

Whether Instructables is too complex or not it is still one of the biggest digital services for maker projects.

3.2.3.6 Internet forums

Internet forums are one of the oldest forms of online collaboration ("Internet Forum" 2015). A forum consists of a number of persistent discussions that forum moderators have created. Each discussion consists of multiple user created conversations (or "threads") on specific topics. Topics contain messages that are structured into one-dimensional lists (as shown in Figure 11) or branching message trees. Many other messaging services such as Facebook, Slack, blogs, and messaging apps essentially have the same kind of tree-like data structure as Internet forums.

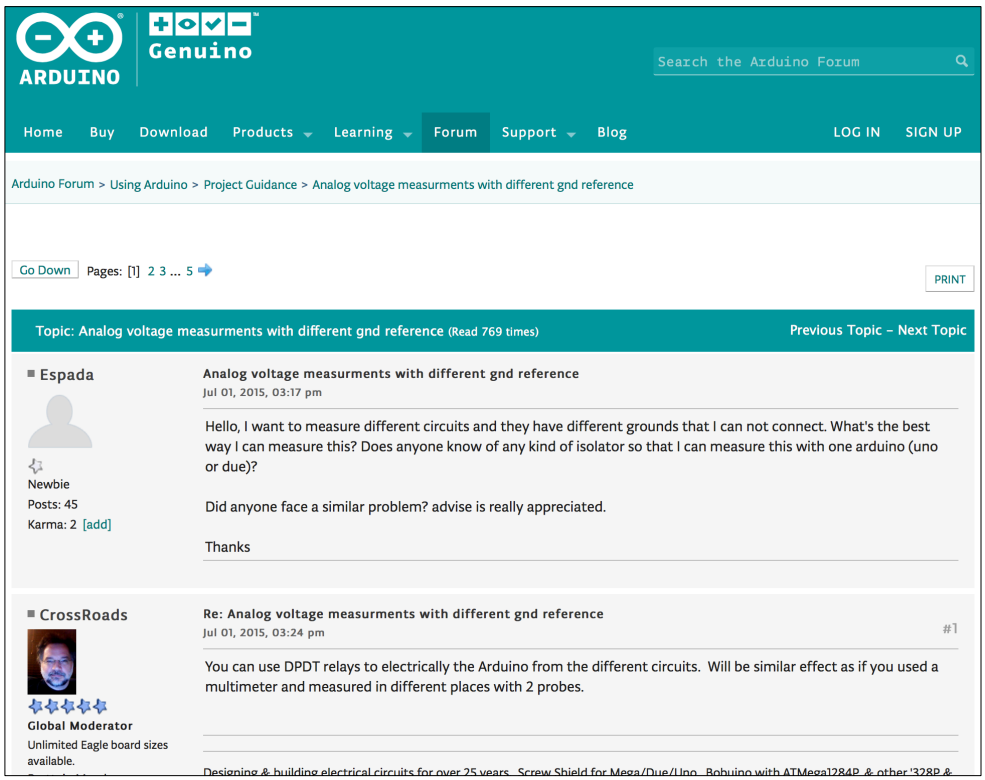


Figure 11: Arduino forum has all the features of a traditional Internet forum.

Most of Internet forums are visible to non-registered users and open for writing to registered users, so all information posted is usable by anyone. Internet forums are extremely widely used for different kinds of knowledge exchange in a wide variety of areas of interest from very specific to very diverse topics including DIY activities (Hyysalo, Juntunen, and Freeman 2013). Many makers use forums to post ideas, find inspiration, ask and answer questions, show progress on their projects, and learn new concepts (Kuznetsov and Paulos 2010).

Openness, accessibility, and familiarity make forums great for makers to share knowledge and their projects, however, discussions in forums are always in flux and it can be difficult to keep track of one's own and other users' projects. In other words old information seems to disappear from view once enough new information comes in and the only way to find that old information is by making a text search. The underlying fact that the forum structure is so simple also plays against it by limiting the usefulness for different kinds of uses than discussions.

3.2.3.7 Knowable

Knowable is a digital service for collaboration and project management for makers and product designers ("Knowable / Polymer" 2015). Knowable allows makers to collaborate on projects by managing teams, hosting and preview files, tracking issues, and getting feedback. Each project has two separate main views, one for showcasing it and one to working on it (Figure 12). It is also used to find new people with the right skills for projects as well as people to find right projects to join. The feature set is similar to GitHub and Instructables but focuses on managing collaborative hardware projects.

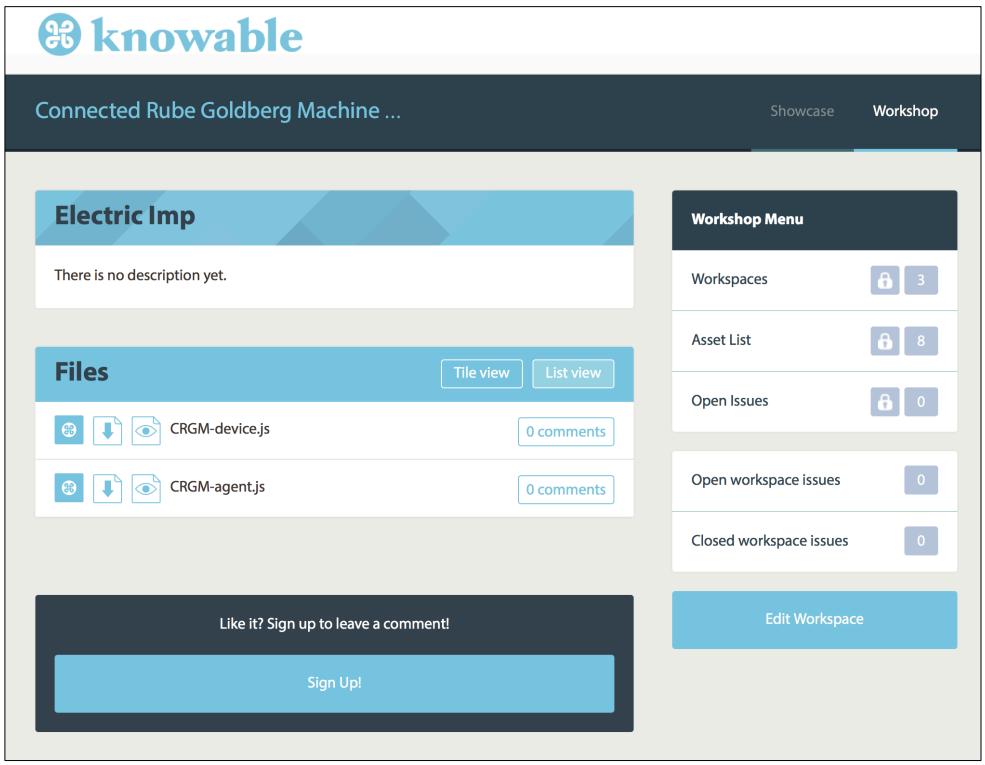


Figure 12: Workshop view of a project on Knowable.

One problem for maker projects is that Knowable is targeted at very specific users – teams of makers who want to use project management tools for collaborating on projects. For many makers Instructables or Thingiverse is a better choice for their fairly simple projects.

According to the founders, the lack of big enough traction the main reason it was shutdown in 2015 ("Spirals." 2015; "Knowable / Polymer" 2015).

3.3 Summary

Despite the difficulties for maker culture based companies to keep openness as a central aspect of their operation after becoming successful, it's nevertheless a sing of success for the maker culture to produce such ventures.

Although the future for this new industrial revolution seems promising, it's unclear how the open nature and transparency of maker culture will survive the eventual commercialization of successful maker projects.

Jenkin et al. sees three possible endgames for the maker culture (Jenkins and Bogost 2015).

1. Nothing changes from the current situation, and making remains a self-serving hobby outside of real-world activity.
2. A growing commercial ecosystem that rises to support maker activities and becoming more and more part of the maker culture.
3. An evolution that graduates maker to professional practice where making is only a means to get a profession and not the aim itself.

Whatever the future holds for the maker culture, it's seems to be clear that open digital services are crucial for it to be successful and thrive.

There are a wide variety of digital services available for maker projects. Practically all of them have some form of collaboration features from simple commenting to very elaborate version control management features. Many of them have some form of documentation possibilities using different kinds of media from text to images to videos. Only some of the popular services specifically target makers.

Many of the existing digital services have lots of great features but at the same time they still seem to lack in some areas. Seems like there is still space for digital services to be useful for makers in new ways.

Here are some of the most important findings from the benchmark:

- Internet Forums work well even after years of only slight changes to the basic functionality.
- Unnecessary structural complexity is confusing as is apparent with Instructables.
- One of the key aspects of successful digital services seems to be simplicity.
- Services targeted specifically for makers seem to favor polished non-discussion content rather than unpolished content. An exception to this is GitHub, however, it's used primarily for software development.
- Discussion is the only unpolished content that seems to be part of almost every digital service makers' use.

4 Interviews and co-design sessions

This chapter contains descriptions, findings, and design directions of expert interviews and co-design sessions. The planning, executing, and processing the gathered data was done together with Koert. Both of the theses interpret this data individually. The combined interpretation is then used to develop the prototype design.

A user research was conducted in order to understand more about the field of study and the target users. 14 experts that are interested in maker culture were interviewed to understand relevant actors and theories involved in the maker culture. The documentation of the results can be found in Appendix A. Each interview lasted from 1 to 2 hours.

Specialization of the interviewees:

- Maker-to-market makerspace creator
- Prototype and experience designer
- Social media researcher
- Maker culture researcher
- Senior social manufacturing researcher
- HCI researcher
- DIY evangelist
- Crafts teacher
- E-learning specialist
- Two hackerspace core members
- Makerspace manager
- Maker event organizer
- Science workshop instructor

Then 5 co-design sessions were organized with one maker participating in each one of them to understand more about what kind of projects, problems, solutions, and practices they have as well as what kind of digital services they use. The documentation of the results can be found in Appendix B.

Participant in the co-design sessions:

- Fablab staff member
- Crafts teacher
- Maker artist
- Hackerspace core member
- A media student

Each co-design session was organized into two phases. First an interview based on a form (A4) with three distinct areas and the relationships between them. These areas are project, documentation, and collaboration. The second phase was designing together which kind of services they would want and need with their projects.

Each co-design session lasted from 2 to 4 hours.

4.1 Findings from expert interviews

These are the key findings and specific elements found from the interviews.

4.1.1 Key findings

Document easily

- Documenting in a form of a diary rather than a polished tutorial.
- Documenting should be extremely easy and effortless.
- Design a continuous cycle where each effort is rewarded.
- Forcing people to document in makerspaces is not really working well.

Safe environment

- Purposefully create a positive atmosphere of low expectations and supporting failure that leads to a low barrier to document projects. This is opposed to glorifying success. A ranking system is an example where more value is put on people who succeed. However, this doesn't mean that informal hierarchies shouldn't and couldn't happen.
- Join a group of likeminded people. How are values in this group created?
    - *"A system supports the self-esteem of a group by giving status by the amount of shared content. - Marcel Mauss (The Gift)"*
    - Groups need to feel safe.

Openness

- Open source software is valued in the maker culture.
- Project data should be transferrable to other services or be exportable.
- Anonymity helps with creativity but can cause spam problems.

Collaborate

- Also enable groups of people to work on something together.

Work with current digital ecosystem

- Don't make too drastic changes to the current way of working. Also it should work well along side of existing digital services.
- Design for creating new habits, not for breaking old ones.

4.1.2 Potential features

Resulting from analyzing the expert interviews, here is a list of potential elements for the digital service. This list is a precursor for digital service prototype requirements.

- Filter specific types of content in a project view.
- Save a snapshot of a project's current state.
- Get help button for broadcasting a help request to other users. Also have a similar functionality for getting feedback.
- Project overview for people browsing projects.
- Create a summary entry by selecting parts of the previous documentation.
- No specific start phase for a project – can start documenting at any point.

- Linking/referencing to files and code commits.
- In-app browser to help users to save references to external websites and projects.
- Automated patent search for a project.
- Share projects and follow them.
- Share ideas with photographs.
- A goal/manifesto/description of a project/group/individual.
- Use of the system without a login.
- Effortless signup.

4.2 Findings from co-design sessions

These are the key findings and specific elements found from the co-design sessions.

4.2.1 Key findings

Document easily

- Should support the process of learning unlike Thingiverse where you just download a file and print it.

Safe environment

- Design for safe environment without fear of harsh feedback and social rejection.
- Too high quality will decrease motivation to contribute.

Appreciate failure and unpolished content

- Instructables is useful but too high bar to add content.

Collaborate

- Networks should last beyond events and courses.
- Get feedback for projects from other people and experts.

4.2.2 Potential features

Resulting from analyzing the co-design sessions, here is a list of potential elements for the digital service. This list is a precursor for digital service prototype requirements.

- The service should be simple and robust.
- Embeddable to a website.
- Reference other projects.
- Best practices project listing.
- Repeating cycles/steps for project: Maker, Doer, and Thinker.
- Clone/replicate/fork a project.
- Embeddable file previewer.
- Awards, badges and prestige to show progress and motivate use.
- Original “forked” project gains a “father/mother” project status.

- Can follow people and projects.
- "Lurking" or viewing other peoples' projects as anonymous "ghosts" that can be seen by the project owner.
- Avoid long loading and uncomfortable endless scrolling feeds.
- Easy to maintain.

4.3 Design directions

The interviews and co-design sessions show many key findings and here are some of the most important ones.

Document easily
One of the key problems is how to get people to document? Documenting and working at the same time seems to be difficult.

Safe environment: appreciate failure and unpolished content
Why should a maker share their work? What Kuznetsov and Paulos found in their survey and what our interviews and co-design sessions have shown is that only a fraction of the community is making videos and step-by-step instructions, and there seems to be a too high bar to contribute (Kuznetsov and Paulos 2010).

Findings also point out that a digital service for maker projects should be a safe environment where failing is socially acceptable and supported activity. This could mean that content mostly unedited and that polished content is not the norm.

Collaborate
Users should be able to follow projects, each others, and join e.g. groups where they can share their passions.

Openness
The appreciation of open source tools and open source culture could be interpreted as a need to make the digital service into an open source project. Open sourcing a project makes it possible for makers to continue supporting the digital services even if the original digital service should close down.

Open source should also lay a better foundation for data portability or to let users move their data to other services. The public project data could also be open and work with principles of open data in a similar way as DBpedia provides structured data from Wikipedia (Auer et al. 2007).

Work with current digital ecosystem
Makers have an existing digital ecosystem consisting of various digital services and new digital services should work with and fit into this ecosystem rather than try to force or break it.

Makers also have existing habits in the way they work with and use existing digital services. When possible, a new digital service should make use of those existing habits rather than create new ones.

5 Prototype

To evaluate a design for a digital service, a prototype is designed and implemented. A static or animated design relies a lot on imagining the use and creating a mental model in a person's mind whereas a prototype can be experienced more fully when used and only a little is left for imagination (Cardenas-Claros and Gruba 2010). Prototyping also matches well with the iterative and agile development methods used in the production.

Based on the background research, expert interviews and co-design sessions with makers, and on findings in this and Koert's thesis a prototype design was created (Appendix C). The prototype design contains a concept and elements for implementing a digital service for maker projects.

While the detailed description of the design is shown in the Appendix C, a basic usage of the service could go as follows:

A maker is working on a project that she started a moment ago and now wants to show it to her friends. She takes her iPhone and opens Pelori (the prototype mobile application) to capture her progress. She creates a log (a term for a project) and captures few pictures using the toolbar at the bottom of the screen in the project view. Each picture shows up as an entry in the created log similarly to a newly created Facebook post or an Instagram picture. As could be expected other users can comment and see who created the entry and when. She can also write a quick note in the picture entry to clarify what the picture is about. Now that there are some pictures in the log, she first creates a "plan" step entry and then a text entry to explain more what she is going to do next.

Once she has something to show she sends a link to her log to a friend of hers who is interested in the project. Her friend is already a Pelori user and opens up the log by tapping the link she sent. Once her friend now sees her work he feels glad that other people are also struggling with similar problems and that he is not the only one with unfinished projects. He then notices an interesting entry that he is going to use as a reference in his own project. He taps a reference icon below the entry and selects one of his own projects and adds the reference to it. A reference entry is now created to the log he selected.

Here are the main aspects of the prototype:

- Incomplete work is valuable as it shows the process and not just the end result.
- Failure is a good place to learn.
- Easy and effortless to document projects.
- Can show their on-going work to friends, family and other people that are interested in similar projects.
- Have a reference framework that allows tracking interesting projects as well as giving credit to the original ideas, projects, and ways of working.
- Help makers document by providing guidance and light structure with plan, activity, and summary steps.

5.1 Prototype technical design

This chapter describes a technical design used to implement the prototype as well as the whole production. The prototype is planned to be a fully functional Internet connected mobile application for iOS and Android ("Apple - iOS 8" 2015; "Android" 2015). Some of the planned features could not be implemented since the prototype needed to be ready for evaluation at the Maker Faire UK convention. The aim is to evaluate the prototype and use it as a design tool for the beta version.

5.1.1 Requirements

Based on the expert interviews and co-design sessions here are the requirements for the digital service prototype. These requirements take into account what kind of design the prototype has and what kind of technology is currently available.

For the digital service, a front end is needed for the user to interact with the service and a back end to persist and process data. Here are the main requirements for the technology used to implement the digital service.

Fast to develop cycle
Should be able to implement both iOS and Android versions in a fairly limited time (one month).

Cross-platform
The prototype needs to run as a mobile application on iOS 8 or newer and Android 4.3 and newer. Either the beta version or the release version should have a browser version that runs on a desktop computer.

Online
An Internet connection is required whenever the prototype is used.

Native like application experience
The front end should be built as a native like mobile application for a mobile phone.

Modularity
The back end should be designed in a way that it could be used either through a mobile phone or a browser. Neither one should not be restricted by the other.

Mobile back end as service
Back end should be a Mobile Backend as Service (MBaaS) with features such as user management, database, social networking service integration, and push notifications should be available out of the box.

Real-time data syncing
Changes doesn't need to be synced in real-time between users viewing a same page although real-time data syncing would be a good have later on.

Push Notifications
A push notification element doesn't need to be implemented but the back end needs to provide it when required.

Web server
Web server would be good to have later on for the browser version.

Server-side code
Should be possible to run server-side code if required.

Analytics
Analytics should be available if required.

Back end versioning
Built-in support for back end versioning would be good to have since at any point there are multiple versions of the application in use. Thus the back end should have corresponding versions responding to the client's requests. Although it would be good to have multiple versions of the back end running in parallel it is still not a definite requirement since it's still possible to get around by customizing the back end code by making the client send it's version whenever it makes a request to the back end.

5.1.2 Choosing technology

The aim for choosing the technology for the production is twofold. First, it needs to enable fast development cycles since there is a deadline for the prototype. Second, it needs to be future proof to be able to provide services that the beta version and later the release version will require.

5.1.2.1 Front end

The cross-platform requirements combined with a need for fast development cycles and limited resources, in practice, means creating the digital service as a hybrid application (Xanthopoulos and Xinogalos 2013). The use of HTML5 technology in a hybrid application allows faster creation of a browser version since in a hybrid application the UI and the application logic in essence runs in a browser WebView. The de-facto container for hybrid applications is an open source project called Cordova ("Apache Cordova" 2015).

Feature	Ionic	Appgyver	Appcelerator
Builds	CLI (local)	CLI (cloud)	CLI and IDE
iOS support	Yes	Yes	Yes
Android support	Yes	Buggy	Yes
Browser support	Possible	-	Mobile Web
Layout language	HTML	HTML	Alloy
Logic language	JavaScript	JavaScript	JavaScript
Container	Cordova	Cordova	Titanium
Open source	Yes	Yes	Yes

Table 4: A comparison between hybrid mobile application frameworks.

So with the limitation of hybrid applications Table 4 shows a comparison of different hybrid mobile application frameworks.

Ionic
Self-description: "The beautiful, open source front-end SDK for developing amazing mobile apps with web technologies."

Ionic, created in 2013, is a mobile front end SDK for hybrid applications ("Ionic" 2015). Ionic is essentially a JavaScript, HTML and CSS front end library that runs on top of Cordova and makes it easy to build native like user interfaces for various platforms such as iOS and Android. The default and recommended JavaScript framework for Ionic is AngularJS which is an open source project maintained by Google ("AngularJS" 2015).

Ionic excels in quick cross platform development and allows the use of familiar technologies to build native experiences with very few lines of code. Visual customization is done through CSS and logical customization through JavaScript. Using familiar technologies broadens the number of potential programmers whereas programmers with the knowledge to develop pure native applications on both Android and iOS are not as common.

Ionic provides command line interface (CLI) for modifying and building applications locally on developers' computers. Building a project is a fairy quick process taking only 5-10 seconds and doesn't require an Internet connection. Developers are free to effortlessly customize their projects with open source plugins. Adding a plugin usually requires the developer to run a single line in the command line. Here is an example of this:

```
cordova plugin add org.apache.cordova.dialogs
```

The downside of Ionic is that it's completely open source and the nature of open source is that some bugs take time to get fixed. Also being a hybrid application where the UI technology is HTML and JavaScript the performance is not always as good as with native applications.

AppGyver
Self-description: "Build beautiful data-driven apps with real native performance. We've selected the best of what HTML5, native APIs and the top frameworks have to offer. We've then modified, improved and re-imagined them to bring you the best blend possible."

AppGyver, founded in 2011, is a startup that provides three tools for hybrid mobile application development ("AppGyver" 2015).

- Supersonic for building user interfaces that are hybrids between native and HTML5.
- Steroids, a CLI for building, testing, and distributing.
- Composer, a fairly complete browser based tool for bootstrapping applications.

AppGyver is essentially a set of tools built on top of Ionic. The goal is to make it easier to build Ionic applications. Many of the features AppGyver provides are welcome and make developing easier resulting in an application that's performance is closer to a native application rather than a pure Ionic application.

Their offering seems very enticing but the reality is that by using their tools some of the flexibility and freedom of developing a pure Ionic application is lost. Most notably the build process is limited to their cloud build service that is much slower than pure Ionic local build process.

The other downside is that, although the Composer is fully featured, capable, and easy to use, it's still only meant to bootstrap an application. So in practice the Composer is used first and then once a certain level of development is reached the application is downloaded as a Steroids project and the development is continued out of Composer from there on.

Appcelerator
Self-description: "Everything you need to create great, native mobile apps—all from a single JavaScript code base."

Appcelerator, founded in 2006, is a complete platform for mobile application development ("Appcelerator" 2015). Where Ionic has Cordova and Ionic SDK Appcelerator has Titanium and Alloy MCV framework. Although all of these tools use JavaScript and are open sourced Alloy requires developers to learn new HTML and CSS like languages to build user interfaces. Also although open sourced, Titanium and Alloy require contributors to sign a Contributors License Agreement (CLA) before any changes can be contributed to the open source project.

The strength of Appcelerator is that by replacing HTML and CSS layer they have been able to make Alloy compile into native code resulting in native like UI

performance. The resulting application is even more native like than what AppGyver applications are.

There are some benefits to using Appcelerator, however, having to learn a new language always takes a lot of time and it is a big risk in the long run to use technology developed by only one company.

Selected: Ionic
Although Ionic stack is not as fully featured and UI in Ionic applications do not perform as fluidly as in AppGyver and Appcelerator applications, Ionic is selected because of it's openness, tool set flexibility, and full control over the build process.

Appcelerator applications' UI performance is very close to native and although their stack is open sourced, building Appcelerator applications require a developer to learn a new language. In summary, choosing between HTML5 and Alloy is an easy choice to make in favor of HTML5 as a standard hybrid application development technology.

Many of the shortcomings that AppGyver currently has are likely to be remedied in later releases. Also, the tie-in to their tools should be loosened so that it would be possible to continue developing an AppGyver project with standard Cordova and Ionic tools at any point.

5.1.2.2 Back end

Table 5 shows a comparison of main technical features for the production between three popular back end services for mobile cross-platform development.

Feature	App Engine	Parse	Firebase
Service type	PaaS	MBaaS	Baas
Back end Versioning	Yes	-	-
Push Notifications	Difficult	Easy	-
Tasks	Yes	Yes	-
Scalable NoSQL database	NDB	MongoDB	MongoDB
Binary data storage	Yes	Yes	-
Real-time data synchronization	Difficult	-	Easy
Analytics	Good	Good	Basic
Documentation and community support	Great	Great	Good
Web server	Complete	Good	Basic
Social Network Authentication	Complex	Effortless	Effortless

Table 5: Comparison of three back end providers for main technical features for the production.

Following is the rationale for selecting a back end for the production. All of the compared back ends have automatic scaling, NoSQL database, easy deployment, analytics, and generally are automatically managed modern services for hosting Internet connected digital services.

Google App Engine

Self-description: "Run your applications on a fully-managed Platform-as-a-Service (PaaS) using built-in services that make you more productive. Just download the SDK and start building immediately."

Google App Engine, initially released in 2008, is a fully featured and battletested platform as a service for all major platforms ("App Engine" 2015). It runs on Google's infrastructure. It has features for almost everything, but it can be overwhelming and complex. It requires a lot of research and learning to choose the right option from Google's numerous offering. In short, you can do almost anything with it but it requires more work to get started. So getting started and developing a prototype quickly is not that feasible.

Parse
Self-description: “Build your perfect app on any platform. Focus on creating amazing user experiences and forget complex infrastructure. Instantly add a powerful cloud database, push notification services, and analytics tracking to your app. Choose from over thirteen native SDKs for mobile, desktop, and IoT devices and discover all that Parse has to offer.”

Parse, founded in 2011 and acquired by Facebook in 2013, is a Mobile Backend as a service for all major platforms (“Parse” 2015). It is fully featured and has a highly streamlined way developing digital services. In contrast to Google App Engine, there is less room for customization and some of the things can only be done in one way. However, enables a very streamlined and effortless development process.

Parse comes with a good support for JavaScript that can be used to do almost everything the REST API can. The downside is that the Parse JavaScript API uses Backbone.js models and the Ionic framework is based on AngularJS models. However, this mismatch can be remedied by patching the Parse models so that they work with AngularJS (“Brandid/parse-Angular-Patch” 2015).

Firebase
Self-description: “A powerful platform for your mobile or web application. Firebase can power your app's back end, including data storage, user authentication, static hosting, and more. Focus on creating extraordinary user experiences. We'll take care of the rest.”

Firebase, launched in 2012 and acquired by Google in 2014, is a Backend as a service for mobile and web applications (“Firebase” 2015). Unlike Google App Engine and Parse, Firebase is not a fully featured back end for application development but is focused on providing a database that synchronizes data in models across multiple devices in real-time. This feature is highly desirable for services like chat applications that have frequently changing content and is viewed by more than one people at a time.

Although Firebase makes it very easy and quick to start using the database and have synchronized data across users and devices, it crucially lacks in other areas such as not having proper binary data storage for images and missing an integrated Push Notification service.

Selected: Parse
Firebase was dismissed first since it cannot provide all the required features for the production in the long run.

Although Google App Engine is by far the most fully featured and flexible, in the end Parse was selected since having a hard deadline for the prototype requires getting started fairly quickly. One good example for Parse’s benefit when comparing to Google App Engine is the extremely simple process of setting up Push Notifications.

So in summary the front end consists of an HTML5 application that runs on cross-platform native wrapper called Cordova. The HTML5 part consists of Ionic UI framework that uses AngularJS framework as the model-view-controller (MVC) architecture.

5.1.3 Architecture

The architecture should be able to fulfill the requirements laid out in the previous requirements chapter.

To certain extent the architecture needs to be robust enough to withstand fairly big changes that comes with developing digital services with multiple iterations (prototype, beta version, release version).

The back end and front end technologies and platforms should be independent from each other. For example if the front end platform changes from a hybrid application to a native application the back end would require only minor changes.

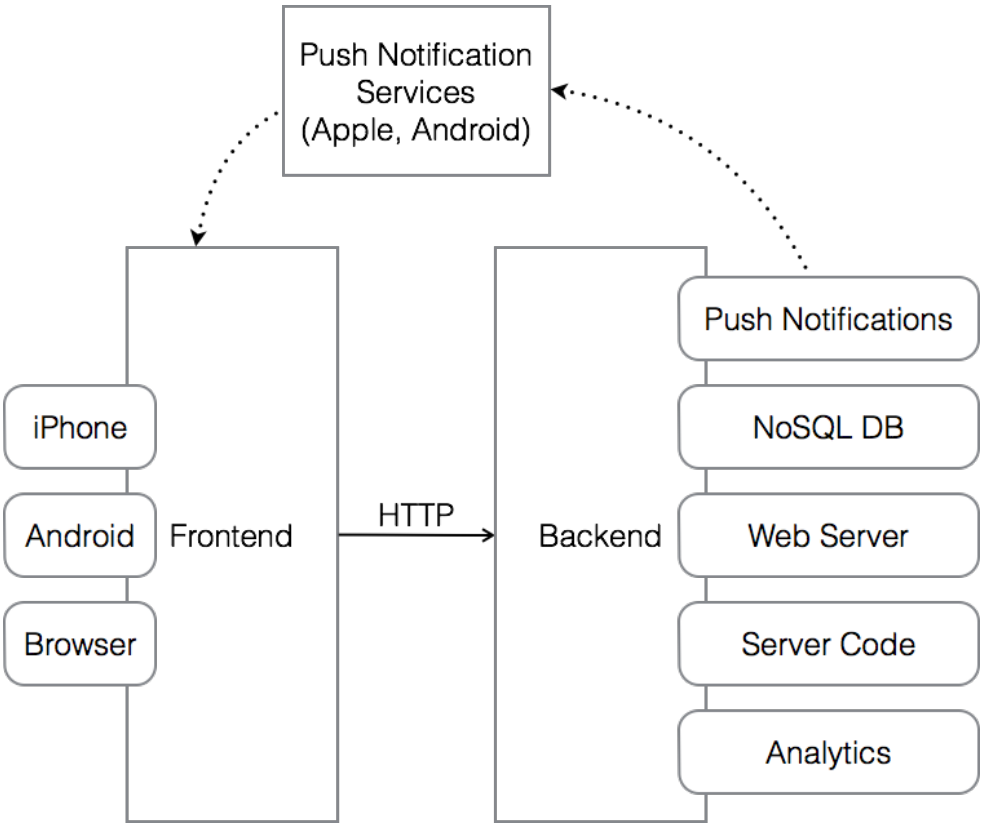


Figure 13: Overall architecture for the digital service.

The overall view of the architecture is shown in Figure 13. More specific architectures of the front end and back end are described in the following two chapters.

5.1.3.1 Front end

Ionic is the selected front end framework for the production and the recommended JavaScript framework for it is AngularJS.

Modified Parse models so called Parse Objects present the data models in the front end. There is a disparity with Parse and Ionic since a Parse Object is an implementation of BackboneJS model that doesn't play well with AngularJS. Since

Ionic is built to work with AngularJS framework and as such is not that compatible with BackboneJS, the default Parse Objects need to be patched in-order to work with AngularJS. This open source patch, further modified for this production to fit Parse Objects better with AngularJS, can be found at: https://github.com/taromorimoto/pelori/tree/master/pelori-app/www/lib/parse-angular-patch.

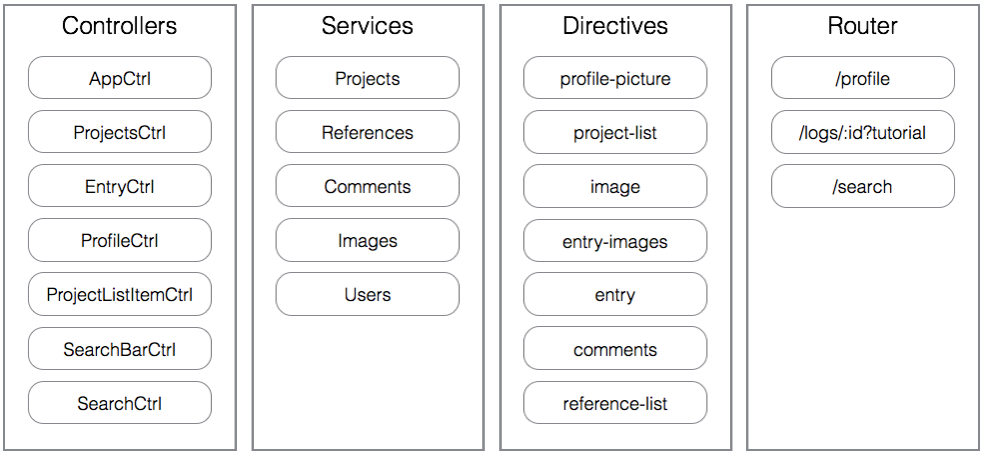


Figure 14: Front end elements adhere to AngularJS framework.

As shown in Figure 14 the front end application logic is divided into 4 different groups of components.

Controllers contain the business logic of the application. Services separate the back end connection logic from the rest of the app. Directives are dynamic extensions to the HTML presentation logic. Router directs traffic from certain URLs to certain HTML templates and controllers.

5.1.3.2 Back end

Parse, used for the back end, provides many of the technologies required for the production. The database, MongoDB, provided by Parse is a NoSQL database that allows developers to build applications that scale easily ("MongoDB" 2015). It stores simple JSON-like data structure that can be used as-is on the client-side ("JSON" 2015). The database models are described in detail in Figure 15.

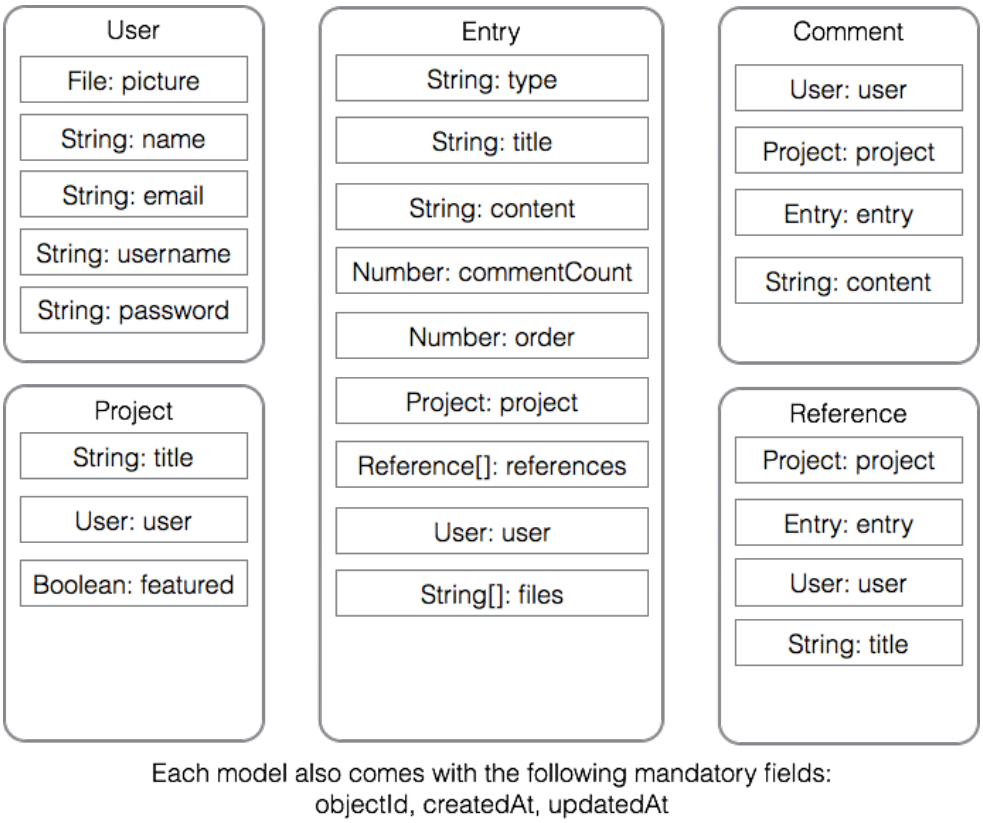


Figure 15: Database models for the prototype.

These database models or Parse Objects are modified through equivalent JavaScript objects. For example a Project model can be first declared and then created and saved on the client-side JavaScript by running the following lines of code. Please note that the declaration is slightly different here since some modifications are required to make Parse Objects compatible with AngularJS.

```
var Project = Parse.Object.extend({
    className: 'Project',
    attrs: ['title', 'user']
})
var project = new Project({
    title: title,
    user: Parse.User.current()
})
return project.save()
```

If there is a need to process some business logic on the server-side for security or some other reason, then this can be achieved with Parse's Cloud Code. In essence it is a modified version of NodeJS that runs JavaScript code on the server ("Node.js" 2015).

Parse also provides an automatic user management through a specific User model. Parse automatically handles sign-up, sign-in, and persistent sessions.

Images and other binary files uploaded by the user can be stored as File objects. On the client-side these objects provide an URL to access the uploaded file over the Internet.

Notifications can be sent through a unified API that Parse provides. Notifications are not implemented for the prototype due to time restrictions.

5.2 Prototype evaluation

The prototype was evaluated at the Maker Faire UK and Media Lab Helsinki Demo Day (Appendix D). Figure 16 shows three screenshots of the implemented prototype.

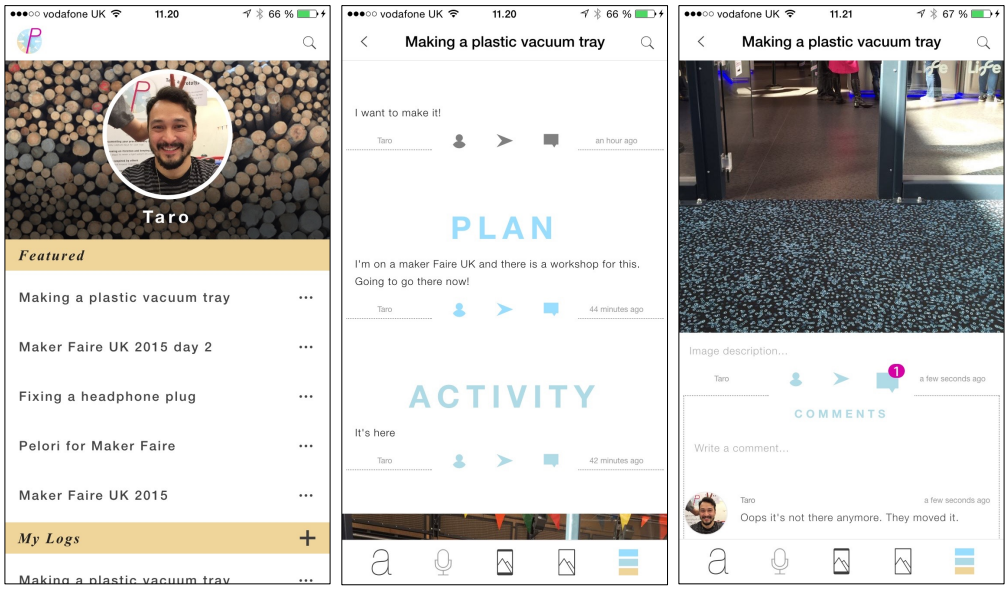


Figure 16: Screenshots of the prototype evaluated at the Maker Faire UK.

The technology chosen for the prototype worked well and it took only a month to develop it. After initially developing a working version for the iPhone it took only some hours to produce a working version for an Android phone.

5.2.1 Key findings

The prototype was evaluated at the two events by pitching and probing. It was used as a communication tool to make it easier for subjects to immediately understand how the service works and how it feels. In practice, it was a demonstration in combination with a 1-20 minute conversation with the subjects. The overall feedback on the prototype was positive and it seems like there is a need for this kind of digital service. We presented the prototype ourselves so this might have contributed to the minimum amount of negative feedback that results from social inhibition.

Document easily

Removing steps:

- Steps create more complexity and in so are getting in the way of documenting.
- Makers have very little time for planning, so concentrate only on making it easy to just add basic content.
- This is good for reducing the code complexity that had to be done to color all the entries that followed a step.

Multi-platform:

- It's great that there is an iOS and Android version.
- Browser version would complement certain use cases.

Tagging/annotating pictures:

- A key feature.
- Annotate a picture by tagging a certain position in a picture and adding a textual/audio comment on it.

Dictation:

- Audio recording and speech to text would make documenting easier.

Toolbar:

- One icon for capturing a photo and getting from photo library.

Moving entries:

- Moving entries between projects in case you create a new subproject from using existing entries.

For teachers and students:

- Easier work for students to document their course work and process.
- Great for teachers to track students progress and give comments on the work.

Safe environment: appreciate failure and unpolished content

Safe to fail:

- There is no specific feature for this but this idea needs to be made clear to the users.

Collaborate

Private version:

- Private version for companies for internal projects and other uses such as travel reports.

Parallel projects:

- Projects can have other projects or subprojects connected to them.

Have pictures in one place:

- Now some teams use messaging apps where pictures tend to be lost after some time.

Openness

Export:

- Possibility to export a project as a PDF or some other format.

Follow projects and people:

- Follow projects and people and get notifications from their activities.

Work with current digital ecosystem

Project timeline:

- It's easy for people to understand the project entry list as a timeline. It makes it easy to understand it as such since it reminds then of social network feeds like Facebook.

Live blogging platform:

- The service could be used as live blogging platforms as its features closely resemble live blogging functionalities.

5.2.2 Technical design directions

For the most part the original technical design done for the prototype should remain mostly unchanged. However there are some specific things that should be changed. The following technical design directions were found.

Removing entries
Entry steps are not really required by the users and they also cause unnecessary technical complexity. Steps should be removed and different types of entries should be treated as just an entry with different kinds of data attached to it.

Richer content
There is a need to add richer content than just pictures. Options for these are videos, audio, and GIFs. All of theses options require the research and evaluation of related JavaScript Cordova plugins and native options.

Replacing entry references
Entry based references turned out to be too specific and tedious to manage. Instead project level connections and entry-level URL links could be a better option. These changes require replacing References database model with something else.

Push notifications
Push notifications should be implemented.

Refactoring code
Code should be refactored so that code for specific functionalities would preside in the same place. For example controllers, directives, and services related to projects should be in "projects.js" file.

6 Beta version

The beta version was implemented in a little over a week. The lack of time meant that many of the planned features didn't get implemented. Overall the service works very similarly to the prototype, however, there are a number of features that got changed and added. This chapter describes the most relevant aspects of the beta version. The detailed description of the planned features for the beta design can be found in Appendix E.

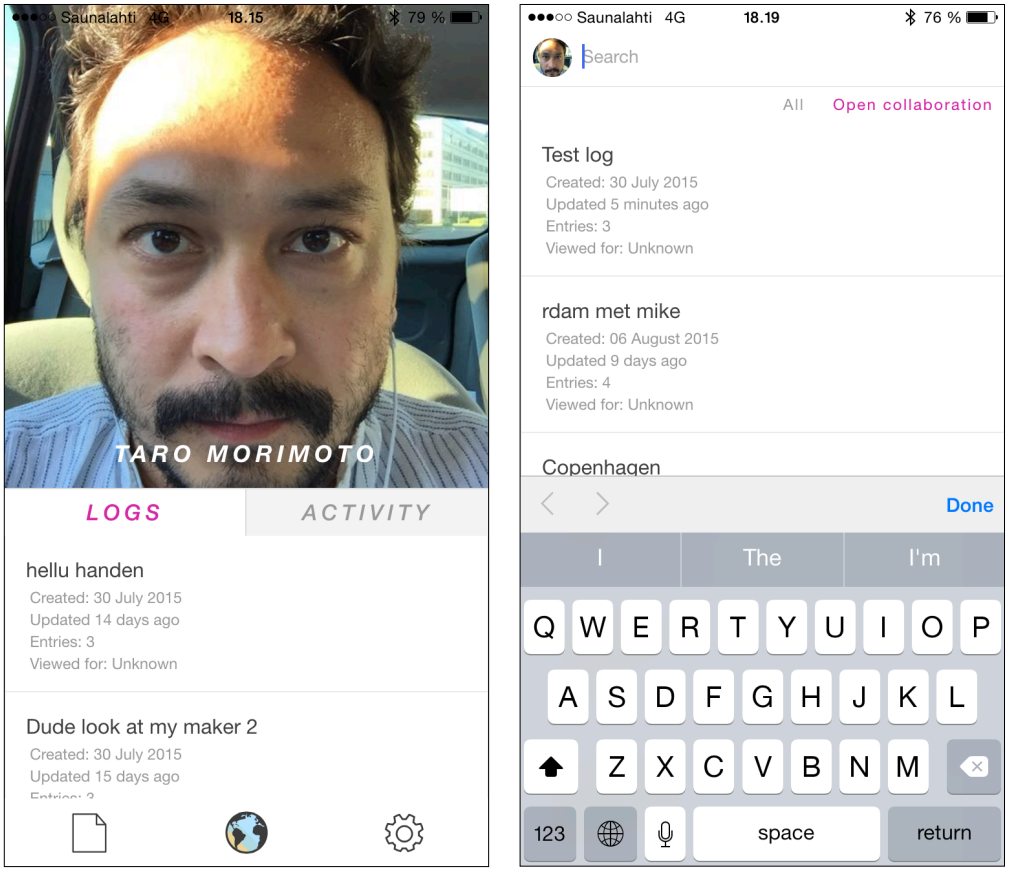


Figure 17: Screenshots of the implemented beta version on iPhone 6 showing the profile picture view on the left and the search view on the right.

The profile page in the beta version, as shown in Figure 17, allows users to view other users profiles. The search view on the right in Figure 17 shows "Open collaboration" that is used only to search the projects that allow all users to add new entries as opposed to the public projects were all users can only make new comments.

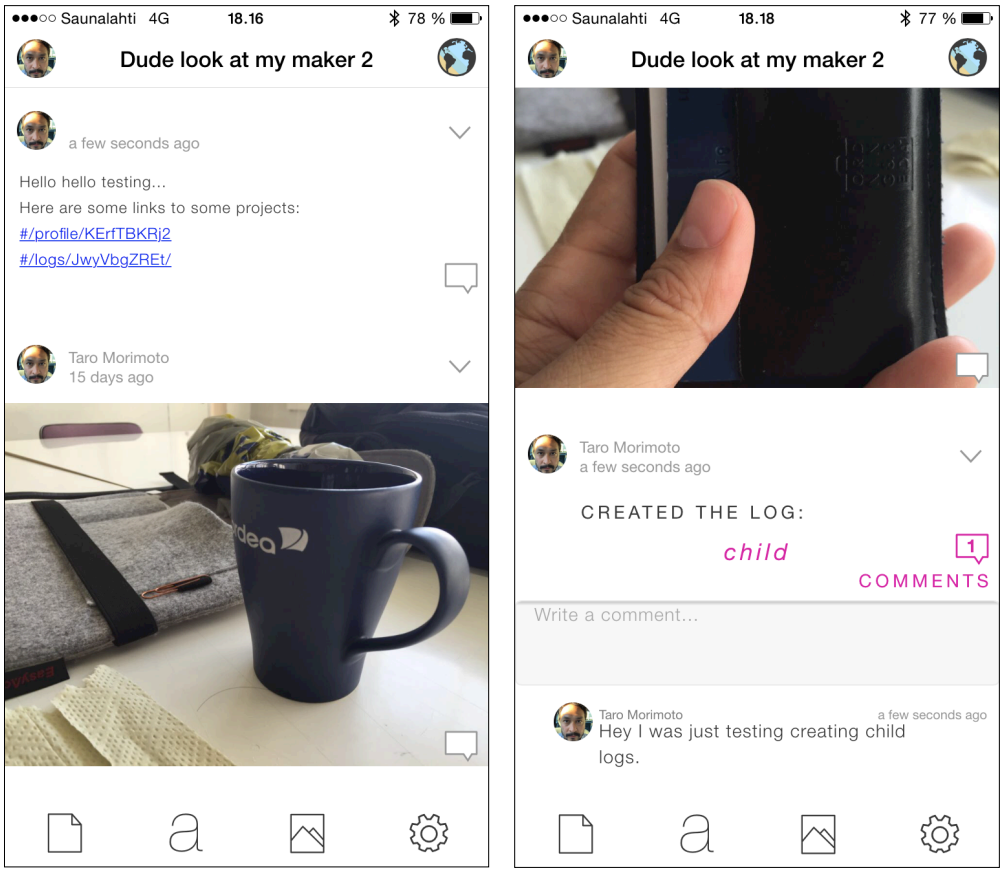


Figure 18: Screenshots of the implemented beta version on iPhone 6 showing the project view on both pictures.

The project view changed quite a bit in the beta version as shown in Figure 18. The toolbar was completely redesigned, steps were removed, and entry based referencing was redesigned to be project based. This project linking is described in more detail in the next chapter.

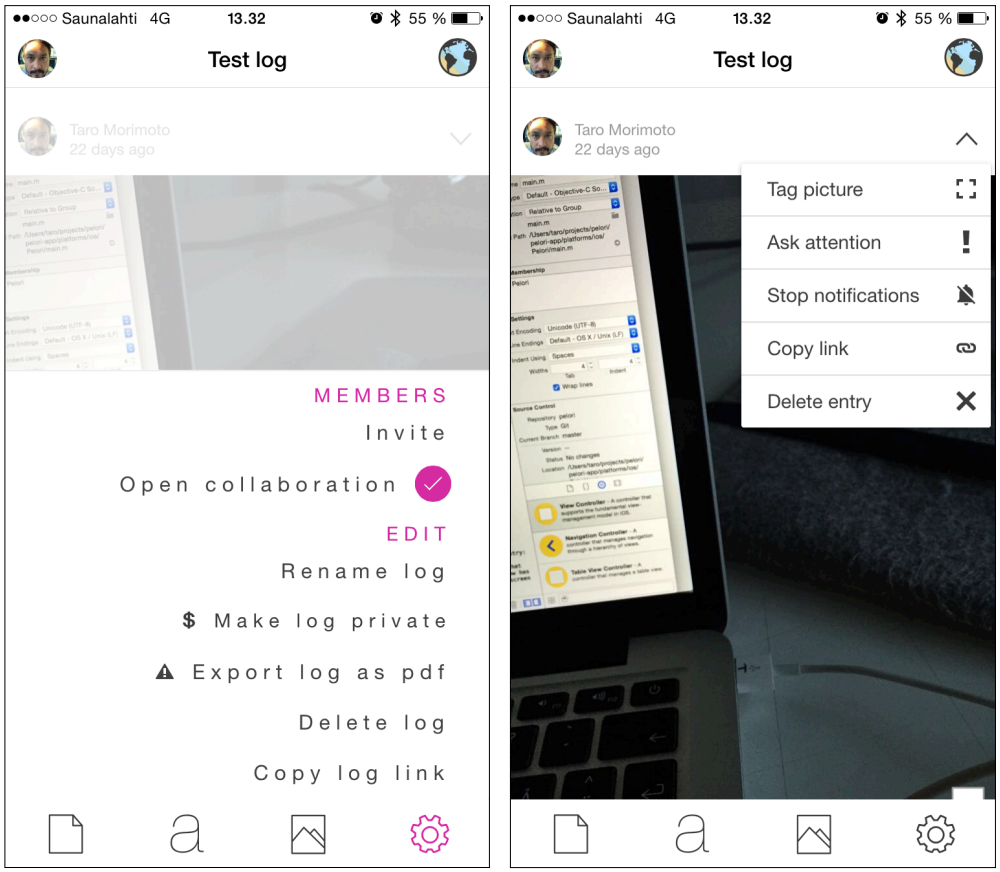


Figure 19: Screenshots of some of the non-functional features in the beta version.

Many features were designed and implemented into the UI as non-functional buttons (Figure 19). Tapping a non-functional button informs the user that the feature is not yet implemented.

- Following projects
- Making projects private
- Exporting projects as PDF
- Inviting collaborators to a project
- Tagging pictures with comments
- Getting and stopping notifications on entry activities
- Asking attention from the community to get help and feedback for an entry

6.1 Technical design

There were no mayor changes to the technology and architecture of the service. The beta version mostly added new functionality but there were also some features that got removed or changed.

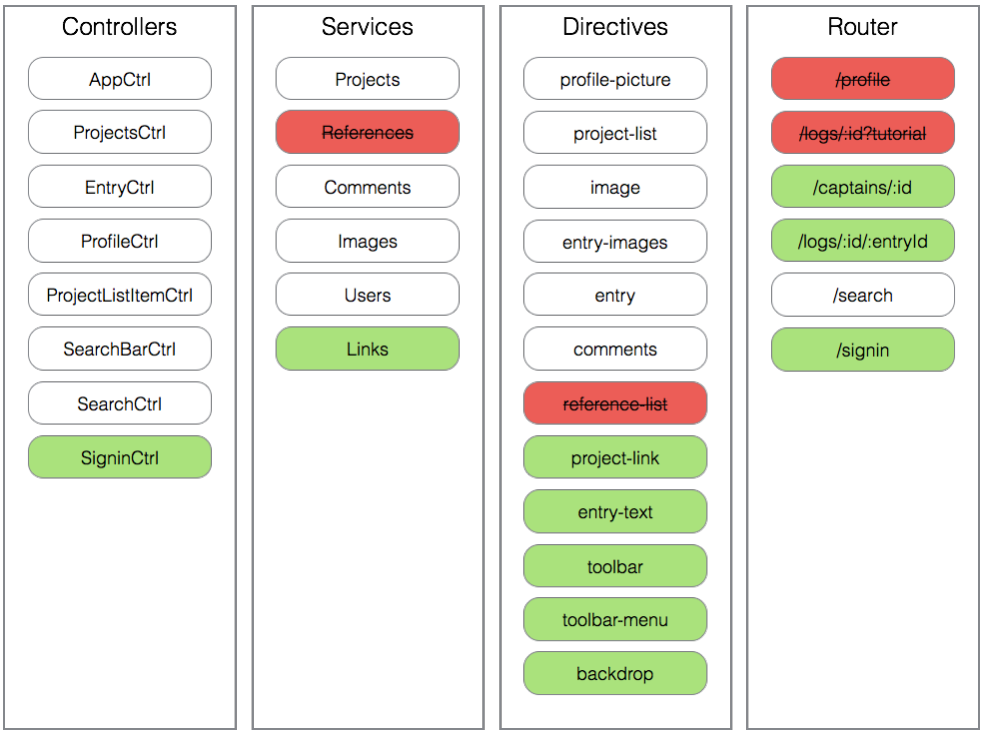


Figure 20: Front end changes for the beta version.

Figure 20 shows the relevant changes to the front end between the prototype and the beta version. The overall structure of the services, controllers, and directives remained unchanged but some of them got removed and some got added. Some things also changed like the URL for the profile page from "/profile" to "/captains" conveying a playfulness when used on the browser where the URLs are visible.

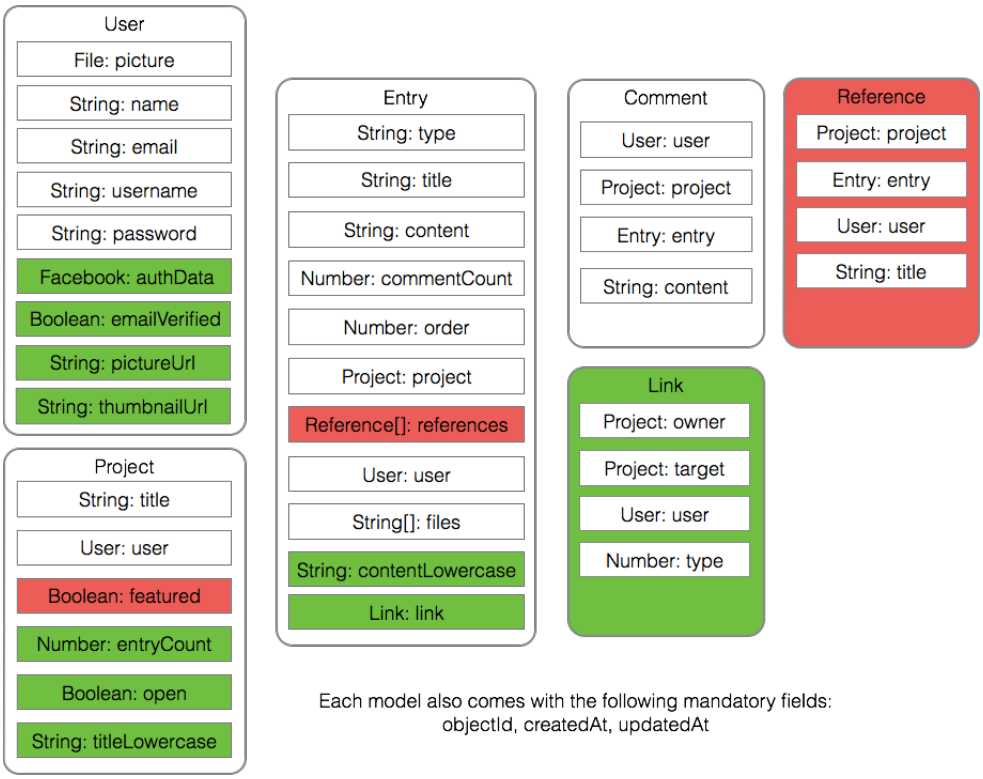


Figure 21: Back end changes for the beta version.

The backend changed a bit more than the front end. Figure 21 shows the relevant changes to the back end between the prototype and the beta version. The biggest change was the removal of entry-based references with a more generic Link model that is used in the beta version for project based referencing.

Following is a description of the most significant changes and new features for the beta version.

GIF animations

Some effort went into researching and implementing a support for GIF animations in entries and profile pictures using an open source JavaScript library to create GIF animations from video ("Gifshot" 2015). Although there was some progress in managing to implement a working version for Android, after one day of work it was obvious that there was too little time to make it work on iPhones.

The main reason was the lack of support for WebRTC on iOS ("WebRTC" 2015). An open source plugin that was created to implement the WebRTC API on iOS was also used to make it work on iOS, however, the codec in captured video was not compatible with the GIF JavaScript library that was used to compile the animated GIF ("Cordova-Plugin-Iosrtc" 2015). It's possible to capture video and create animated GIFs but until the same codecs are supported this feature cannot be used easily in hybrid mobile applications.

Crosswalk for Android

Currently the Android platform is very fragmented and it's causing a lot of problems for developers and the operating system ("How Fragmentation Affects

the Android Ecosystem - TechRepublic" 2015). The problem is that there are many different browser versions that Cordova is running on and this makes it difficult for developers to know on how many Android phones an application truly work. To remedy this, Crosswalk for Android has been created to have the same exact browser version in a Cordova application ("Crosswalk - Build World Class Hybrid Apps" 2015). This also allows capturing GIF animations to work on Android phones where the default browser wouldn't normally support it.

Facebook authentication

Email authentication was replaced with Facebook authentication due to the challenges in creating an instant and reliable email sending service in a sort amount of time. Facebook authentication also allows a smaller amount of steps to start using the service.

Project linking

Connecting projects together to create meaningful collections of projects was implemented using a Link model (Figure 22). Using a link object with a type allows various kinds of connections between projects to be established including sub-projects and project references. Links can also be used in situations where there is only one project involved such as favorite projects, following, subscribing to notifications, and other activity that could be shown on an activity feed.

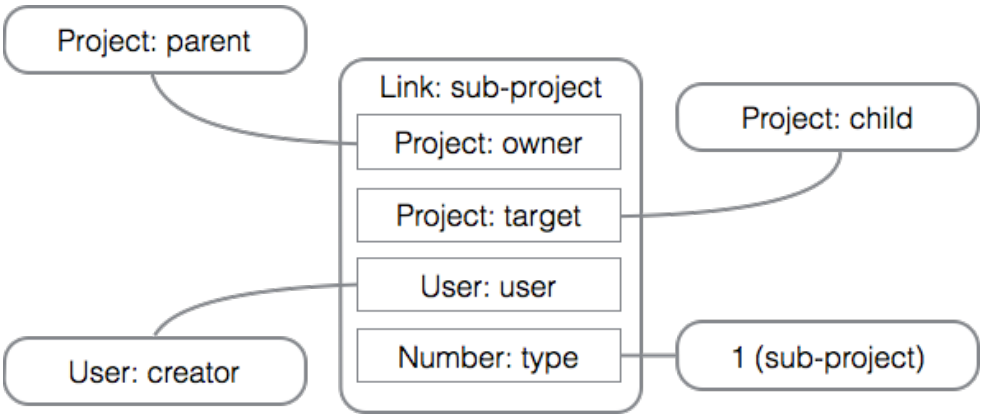


Figure 22: Example of project linking using a Link model.

Rich comments

Comment model was to be removed and replaced with tree like Entry structure where a comment would be an Entry model with a pointer to a parent Entry. This would allow for richer content to be included into the comments mostly using the same code that was used to create top-level entries. Due to lack of time this feature was not implemented.

Case insensitive search

Case insensitive search was implemented by creating a lowercase copy of searchable text every time an entry or project was saved to the server. Comments and users are not yet searchable.

Toolbar

The toolbar at the bottom of the project view was redesigned so that it is now visible also on the profile page. The content and use of the toolbar varies depending on the context so that it looks and behaves differently depending on what page the user is viewing and what kind of permissions the user has to the viewed content.

6.2 Beta version evaluation

The original plan was to benchmark the beta version against a limited number of similar digital services being used for maker projects. However, due to the lack of time to properly implement the beta version the current version is missing many of the key features that are required to truly stand on its own against the competition. The current beta version also suffers from apparent bugs. In a way there are 2 beta versions of the service; one that is implemented and one that is only designed but cannot really be tested.

Here is the list of key features the implemented beta would need.

- Browser version for faster writing and searching.
- Rich comments with pictures, audio, and GIFs.
- Following users and projects.
- Subscribing to entry activity to get push notifications.
- Annotating pictures though when commenting on pictures.

As a result it is not useful and would be a waste of time to benchmark the current beta version against the competition.

However, the designed beta version seems to answer some aspects that the competition is not catering to. **Pelori could be a place for persistent, continuous, and evolving maker projects where other people could follow, comment, and contribute to on-going projects.** Some of the features that would enable this kind of a service already exist in competing services; however, it seems that there aren't any digital services that would provide exactly that.

GitHub has all these features but it's too granular and is geared towards software development. Instructables has a large community of makers and vast amounts of content in the form of tutorials. These polished tutorials are a great source of knowledge for the makers but they only show the end result, not the journey that contains so much more information. Internet forums are a place where these journeys are exposed to other makers; however, they only serve to show glimpses into projects. There aren't any mechanisms to track projects. The discussion, although highly important, is in the center of Internet forums not the projects.

This is where the designed beta version of Pelori could provide something currently missing from the makers' digital ecosystems.

7 Discussion

Teamwork

I started to work on my thesis alone more than a year ago. After many months of work, Koert and I decided to make a joint thesis project. It has become clear to me that working with someone on this kind of a project is a great source of motivation, inspiration, and new ideas. Rather than working alone, where getting feedback on your work is always a bit tricky, working in a team gives you the possibility to always ask for a second opinion.

A joint master's thesis

Currently at Aalto University it seems extremely uncommon to make a fully joint master's thesis with one resulting book where some of the content in the body of the thesis is common and some individual. Based on the experience I've had making a joint thesis at Aalto University, I believe it would be good for students if a format for a joint master's thesis would exist as an option to choose from since some joint thesis projects can be more difficult to split that others.

Even if such a format would exist there would still be many downsides to it such as difficulties in writing common texts where every sentence needs to be discussed together and in doing so slowing down progress. One other major downside is that evaluators could find it laborious to keep track of who wrote what.

Our original plan was to have common and individual parts in a thesis fully joint thesis but in the end we ended up making two completely separate theses with only a common production. For us making separate individual theses' worked, however, now thinking back it's not clear if one way would be better than the other.

Design through research

Having worked as a professional programmer for the past 14 years I have observed that an idea often emerges from the business side of a company and is then worked on by a team of employees to develop the design. In a good case this involves some effort to investigate target users but is usually left out since the product owner has worked with the users and similar products for some time. The experience we've had with this thesis project is that doing user and literature research seems to benefit the design and the overall understanding of what is important and relevant to the users and the practice and what isn't.

Once there is enough understanding of what to do and how, much of the guesswork goes away and thus the implementation time actually shortens. This is also shown in this production as the prototype was implemented in one month and the beta version in a little over a week. Both the prototype and beta version functioned well enough on mid range Android and iPhones.

My personal takeaway from this is that the software industry should use user research and literature research more as tools to make better products. Although research takes time and effort, however, the overall development time will most

likely be shorter since there will be fewer iterations needed. In other words, a better design for a prototype can be made before implementation is started.

It should be noted that due to the lack of time our production failed to make non-functional prototypes before a fully functional prototype.

Overlapping iterations

The production had four major iterations: literature research, user research, prototype, and beta version. Looking at the plan it seems like the different iterations were clearly separate, however, in practice some of the iterations crossed over to other iterations and the borders between them became blurred. One of the clearest examples of this was the literature review that in practice spanned across three of the first iterations if not the whole production.

My personal view is that it allowed us to constantly keep up with how other experts have dealt with related issues that came up during user research, prototype development, and prototype evaluation.

But is it really true that overlapping iterations is beneficial for a production or will it only create too much confusion? This might also depend on many factors such as the size of the production team, time span of the production, and the type of product or service in question.

There are many things happening in the maker culture and digital services are one of the key components. I believe how these services are used and what kind of services should be created for makers are some of the things that need further research.

The final design

Will the created app encourage users to produce unedited and unpolished content that will eventually create a messy landfill of unreadable projects? Will it deter users from looking at projects if the content is not of good quality? These are the kind of questions that will get answered once real people use the service for a long enough period of time.

This production was finished when the beta version was created. We will certainly continue developing the service so in that sense there is no final design. We didn't have time to open source the project yet but we have already decided to do it once we have the time.

Future research

In Pelori we ended up with an architecture that was closely related to the way Internet forums work. What we did differently was the way project links allow the creation of a project graph rather than a tree like structure that traditional Internet forums have. Will this kind of project graph work for maker projects or for any other type of users is one of the questions that would be interesting to get more answers to.

One of the key aspects in Pelori is opening up unfinished projects. Are people ready and willing to show their flaws and content that is unpolished? There are many aspects that need more research.

Other areas of use

Although Pelori has been designed for the maker culture, we also designed it to be used by various other users like educators and students as well as companies. One of the first users for Pelori could actually come from teachers using it for a course as a tool to help students document their progress and get contextual feedback from their teacher. If the use continues from course to course Pelori also serves as a tool to use previous work as an inspiration or as a basis for new designs.

8 Conclusion

This thesis set out to learn more about the maker culture and the digital services makers use for their projects. The research question in essence was what kind of digital services can be created so that they will be useful for maker projects. If one could answer such a question it would be easier to make a lot of successful digital services, however, as it's commonly know such formula doesn't exist. I believe the common understanding is that it's almost impossible to know exactly how to design successful digital services thus answers to my research question remain fairly open and inconclusive.

The true test for a digital service is it's launch where it either gets traction or doesn't. There are also many other aspects involved in traction than just the design and execution such as how effectively and in what way the potential users find out about the service and actually keep using it. One could argue that traction has more to do with marketing than good design but I believe these things are mostly a matter of opinion.

We used relevant literature to gain a better understanding of the work other people have do in relation to what we wanted to achieve in the production. User research helped us develop a prototype design that tried to answer the needs of the target users. As using the design we implemented a fully functional prototype and a beta version of Pelori that served well in helping us to further improve the design.

What we kept finding throughout the production of Pelori and our individual theses are the main elements that seem to be important for a digital service for maker projects. These elements are part of the design of Pelori. As we found out through making a prototype some of the elements we initially thought to be important were then discarded or modified for the beta version. The same will most likely happen to the beta version when it will be redesigned for the public launch. The main elements in Pelori are the following.

- Open ongoing maker projects for other makers to follow. This is the main element, and remains to be seen if makers subscribe to this kind of transparent way of working.
- The idea that incomplete is valuable and failing is something to be learned from.
- An effortless way to document projects.
- Simple discussion structure that borrows from existing and successful formats such as Internet forums and Facebook.
- Open collaboration on shared projects.
- Linking projects together.

Starting the project over a year ago, my personal motivation was to build a digital service that would make it possible to share and develop ones own and others' ideas and find inspiration. Throughout the whole process the design of the digital service got more specific and the understanding of what is important became clearer. As a result the design directions for Pelori are now clearer than ever. I hope the future versions of Pelori will help me, makers, and other people to share their ideas and the work that is currently hidden from view.

Bibliography

Ames, Morgan G., Jeffrey Bardzell, Shaowen Bardzell, Silvia Lindtner, David A. Mellis, and Daniela K. Rosner. 2014. “Making Cultures: Empowerment, Participation, and Democracy - or Not?” In , 1087–92. ACM Press. doi:10.1145/2559206.2579405.

“A Movement in the Making.” 2014. Deloitte.

Anderson, Chris. 2012. *Makers: The New Industrial Revolution*. 1st edition. London: Random House Business Books.

“Android.” 2015. Accessed August 21. https://www.android.com/.

“AngularJS.” 2015. Accessed July 23. https://angularjs.org/.

“Apache Cordova.” 2015. Accessed July 23. https://cordova.apache.org/.

“Appcelerator.” 2015. Accessed July 23. http://www.appcelerator.com/.

“App Engine.” 2015. Accessed July 23. https://cloud.google.com/appengine/.

“AppGyver.” 2015. Accessed July 23. http://www.appgyver.com/.

“Apple - iOS 8.” 2015. Accessed August 21. http://www.apple.com/ios/.

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. “DBpedia: A Nucleus for a Web of Open Data.” In *The Semantic Web*, edited by Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, et al., 4825:722–35. Berlin, Heidelberg: Springer Berlin Heidelberg. http://link.springer.com/10.1007/978-3-540-76298-0_52.

Baldwin, Carliss Y., and Eric A. Von Hippel. 2010. “Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation.” *Harvard Business School Finance Working Paper*, no. 10-038: 4764–09.

Blomberg, Jeanette L., and Austin Henderson. 1990. “Reflections on Participatory Design: Lessons from the Trillium Experience.” In , 353–60. ACM Press. doi:10.1145/97243.97307.

“Brandid/parse-Angular-Patch.” 2015. Accessed August 21. https://github.com/brandid/parse-angular-patch.

Cardenas-Claros, Monica S., and Paul A. Gruba. 2010. “Bridging CALL & HCI: Input from Participatory Design.” *CALICO Journal* 27 (3): 576–91.

“Cordova-Plugin-Iosrtc.” 2015. Accessed August 15. https://github.com/eface2face/cordova-plugin-iosrtc.

“Crosswalk - Build World Class Hybrid Apps.” 2015. Accessed August 21. https://crosswalk-project.org/.

“Evernote.” 2015. Accessed July 2. https://evernote.com/.

“Facebook Groups.” 2015. Accessed July 2. http://www.facebookgroups.com/.

Fallman, Daniel. 2009. “Why Research-Oriented Design Isn’t Design-Oriented Research.” *Nordes* 0 (1). http://www.nordes.org/opj/index.php/n13/article/view/222.

“Firebase.” 2015. Accessed July 23. https://www.firebase.com/.

“Gifshot.” 2015. Accessed August 15. https://github.com/yahoo/gifshot.

“Git.” 2015. Accessed July 6. https://git-scm.com/.

“GitHub.” 2015. Accessed July 2. https://github.com/.

“GitHub: Pelori.” 2015. Accessed June 25. https://github.com/taromorimoto/pelori.

“Google Docs.” 2015. Accessed July 2. https://docs.google.com.

Grimme, Shannon, Jeffrey Bardzell, and Shaowen Bardzell. 2014. “‘We’ve Conquered Dark’: Shedding Light on Empowerment in Critical Making.” In , 431–40. ACM Press. doi:10.1145/2639189.2641204.

"How Fragmentation Affects the Android Ecosystem - TechRepublic." 2015. Accessed August 21. http://www.techrepublic.com/article/how-fragmentation-affects-the-android-ecosystem/.
Hyysalo, Sampsa, Jouni K. Juntunen, and Stephanie Freeman. 2013. "Internet Forums and the Rise of the Inventive Energy User." *Science and Technology Studies* 26 (1): 25–51.
"Instructables." 2015. Accessed July 3. http://www.instructables.com/.
"Internet Forum." 2015. Accessed July 9. https://en.wikipedia.org/wiki/Internet_forum.
"Ionic." 2015. Accessed July 23. http://ionicframework.com/.
Jenkins, Tom, and Ian Bogost. 2015. "Escaping the Sandbox: Making and Its Future." In , 29–32. ACM Press. doi:10.1145/2677199.2680558.
Jobse, Koert. 2015. "Untitled Master's Thesis." Unpublished master's thesis, Aalto University.
"JSON." 2015. Accessed August 9. http://json.org/.
"Knowable / Polymer." 2015. Accessed July 16. http://www.knowable.org/.
Kommonen, Kari-Hans. 2013. "Design Ecosystems as the Landscapes for Co-Creation." *Proceedings of the CO-CREATE 2013*, 728.
Kuznetsov, Stacey, and Eric Paulos. 2010. "Rise of the Expert Amateur: DIY Projects, Communities, and Cultures." In . http://dl.acm.org/citation.cfm?id=1868914.
Larman, Craig. 2004. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional.
Leonard, Dorothy, and Sylvia Sensiper. 1998. "The Role of Tacit Knowledge in Group Innovation." *California Management Review* 40 (3): 113.
Lindtner, Silvia, Garnet D. Hertz, and Paul Dourish. 2014. "Emerging Sites of HCI Innovation: Hackerspaces, Hardware Startups & Incubators." In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, 439–48. ACM. http://dl.acm.org/citation.cfm?id=2557132.
Määttä, Anu, and Peter Troxler. 2011. "Developing Open & Distributed Tools for Fablab Project Documentation." In *OKCon*. http://ceur-ws.org/Vol-739/paper_21.pdf.
"Make." 2015. Accessed August 24. http://makezine.com/.
"Maker Faire." 2015. *Maker Faire*. Accessed August 24. http://makerfaire.com/.
"Manifesto for Agile Software Development." 2015. Accessed July 15. http://agilemanifesto.org/.
"Markdown." 2015. Accessed July 6. https://en.wikipedia.org/wiki/Markdown.
Meyer, Peter B. 2003. "Episodes of Collective Invention." *US Bureau of Labor Statistics Working Paper*, no. 368. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=466880.
"MongoDB." 2015. Accessed August 9. https://www.mongodb.com/.
Mota, Catarina. 2011. "The Rise of Personal Fabrication." In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 279–88. ACM. http://dl.acm.org/citation.cfm?id=2069665.
"Node.js." 2015. Accessed August 9. https://nodejs.org/.
"Parse." 2015. Accessed July 23. https://www.parse.com/.
"Pixelapse." 2015. Accessed July 6. https://www.pixelapse.com/.
Polanyi, Michael. 1966. *The Tacit Dimension*. Garden City, N.Y.: Doubleday.
"Press · GitHub." 2015. Accessed July 6. https://github.com/about/press.
"Quirky." 2015. Accessed July 2. https://www.quirky.com/.
Raasch, Christina, Cornelius Herstatt, and Kerstin Balka. 2009. "On the Open Design of Tangible Goods." *R&D Management* 39 (4): 382–93. doi:10.1111/j.1467-9310.2009.00567.x.

"README." 2015. Accessed July 6. https://en.wikipedia.org/wiki/README.
Spinuzzi, Clay. 2005. "The Methodology of Participatory Design." *Technical Communication* 52 (2): 163–74.
"Spirals." 2015. Accessed April 3. http://simonhoeher.com/post/114629725011/spirals.
"Statista." 2015. Accessed July 3. http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/.
"Trello." 2015. Accessed July 2.
Troxler, Peter, and Harmen Zijp. 2013. "A Next Step Towards FabML: A Narrative for Knowledge Sharing Use Cases in Fab Labs." In *International Fab Lab Association, the 9th International Fab Lab Conference, Fab*. Vol. 9. http://fablabinternational.org/system/resources/BAhbBlsHOgZmSSItMjAxNC8wNS8zMC8xNi8xNy81NC83NDIvVHJveGxlcl9aaWpwLnBkZgY6BkVU/Troxler%20Zijp.pdf.
Tsoukas, Haridimos. 2005. "Do We Really Understand Tacit Knowledge?" *Managing Knowledge: An Essential Reader*, 107–26.
"WebRTC." 2015. Accessed August 15. http://www.w3.org/TR/webrtc/.
Xanthopoulos, Spyros, and Stelios Xinogalos. 2013. "A Comparative Analysis of Cross-Platform Development Approaches for Mobile Applications." In , 213. ACM Press. doi:10.1145/2490257.2490292.

APPENDIX

Appendix A. Expert interviews

Co-owner of Maker to Market Makerspace

THE SERVICE	They offer: Try to provide support network from beginning to market (retail space) Street level co working space: cafe, invite people in Interaction with customer (cafe) • Quickly get feedback, questions, comments • Cafe customers, store customers, designers • Made in Finland Space • Shared tools not there anymore • Culture of sharing own tools for other people o Expectation that it's ok to ask o You should say yes. :) • Same price than other co-working spaces • 300-500 euros depending for size • Turn around (people in the space) o 2 ways to leave: grow or die • Not quite like incubator: o Not actively mentoring, more like space where mentoring and help happens • Business model canvas, not done here Events • Started with events before the actual space • Block parties in Punavuori, Kallio o Just temporary, so wanted more permanent solution o Physical presence in the neighborhood
STAKEHOLDERS	Cooperative • Not compulsory • Billing through the cooperative, no need for own company • More than 7 people in cooperative, then can be considered as unemployed o Regular employee

USE CASES	Collaboration • FB group (private) • More meeting before, now less • Also collaborate with past partners before Made in Kallio • Network of references
VALUES	How could they use it: • Sharing is helpful • Clear way of indicating the way things should be used (and open source nature)

Senior UX designer, Prototype design

THE SERVICE	• Service should remind me to
	  • Different aspects of the project grow at the same time
	• Remind to keep as a one package
	  • Parallel tracks
	• Need to make a mark when something is working
	  • Lock a part that is working (history so far)
	  • Video and code combined
	  • SAFE PLACE revert to
	
	Filtering inside large projects would also be fine for him. It would be very nice to see only pictures or errors or code etc.
	
	Toggling between everything and steps is really nice. Scrolling fast through steps might be fast. Steps are needed or at least popular.
	
	“Save me! Save my file!” “Preserving and getting help”
	
	Parallel lines help with opening up parts of a project that might be useful. The entire thing is rarely used completely.
	
	Selecting parts of the complete documentation to create a summary. But he is not doing this at the same time.

STAKEHOLDERS	Hunch: Service might not be for beginners, but for experienced people **with a clearer goal**. Because still these people are not good at documentation.
USE CASES	What is working • Software, code, schematic, hardware • Document what is working: show how it’s working, only the show the code • Need to convey a clear picture: visuals, picture, and videos. Only after that details • Painful to document, but important Overarching goals against sub goals. At the end of … copy

	paste into etc. save People are not documenting in much detail, but do document the simple actions like buying stuff are mentioned on Facebook for instance. Often there is a batch of documentation at the end From a starting point to a saving point. Do people want to even start? Probably he will start with the saving first, and then he can backwards engineer/input the beginning part. Allow people to start into the middle part of a project.
VALUES	It is no problem if the documentation is huge, as long as it guarantees safe places for directions. Links to commits/references to files.

Social psychologist, Social media and IT researcher

THE SERVICE	Profiles: Inside service vs. distributed (selection of sharing contributions) Connected to LinkedIn e.g. for professional Unveil/Reveal comments/contributions to your profile Profile as monitoring your progress Twitter style might push more people to edit but might the standards might become too high □ is there a limit? If the lead users create the atmosphere for "low standards" than the personal restrictions will be diminished. **Ghosts (anonymity and fake identities, why and how?)** Choosing to reveal/connect content to yourself But is failure good? Anonymity gives creativity Interactions with the ghosts (to moderate spam for instance) **Badges/Pins (Hierarchy in between profiles)** Ghosts as a way to break the status quo. Different levels of quality? We need the low quality content to enter the field!
STAKEHOLDERS	• **People waiting for new projects/peer-support** As stimulating you to share your notes
USE CASES	• **How does this differ from rigid/heavy machinery pushing you to document** Reporting for other purposes than other goals than proving actions. Individual goals and community • **Editing as reflecting** Preparing for presentation/broadcasting/publishing. How does the app support the right way of documenting "They are waiting for your notes" Help me!/Demand feedback/Fail or Win/like or dislike

VALUES	• **Removing projects/ or rather pulling them of your desk** Creating visual space Support in failure (atmosphere) Not only glorify success Redefine failure/create supportive atmosphere (Learning from failure) Failing ratio • **Hesitation/Shame of showing incomplete work** = Pain point Could be answered with reflections Problems are too big to be solved in one masters thesis • **Making notes** = Pain point What is the trade off of the sociality ☐ what do you get and what do you lose

Maker Culture Researcher

THE SERVICE	Self selecting/ranking based on your level "Is this review helpful?" feature Ranking problem: Balance between self selecting and others Overview view like for page layout designers Relevance to real world analogues Cycle could be next page/commit/etc.
STAKEHOLDERS	Meritocracy (informal hierarchy) Fab10 big meeting and now putting together a ecosystem of digital tools
USE CASES	Sharing does not happen in Fab Lab Clean vs. messy: control or freedom **Hot, trending, fresh (active users vs. new users)**
VALUES	• "Envision this and that" • "Redefine failing!" • "Fail better", "Do it yourself, but let's fail together!"

Senior Fellow, leading social manufacturing research project

THE SERVICE	Rethink the referencing, include in app browser SCRUM: Did + New plan + Obstacles ☐ we have 2 user roles ACTOR + PROBLEMSOLVER Co-creation web service for multiple companies to sell customized products to customers. Existing already, one company in UK.

STAKEHOLDERS	Obama "Open the research facilities for the makers" Governments (US +EU) are referring to makers on commission level, so it is trustworthy to work with this target group ☐ find references Consumers, Production companies Three parties in designing, making and buying. 1. Makers who design the things and can collaborate a. Basically contains the old design of the service 2. Consumers who can buy the designs 3. Producers who manufacture the designs and bid to produce
USE CASES	User involvement on all stages of the industrial process US Patent Office + Google Patent search ☐ connect with projects Two ways of dealing with patents: • Create something that can be protected • if it is not patentable it means that you (and everyone else) can make it Individuals join together to do/design something. But they can form a group. It should be transferrable and not closed into the service.
VALUES	Bulk production cannot continue to exist, mass customization is the new deal, and supply chain management is the key. Supply should fit the demand.

Feedback after presentation UI Class

THE SERVICE	
STAKEHOLDERS	
USE CASES	Social/Peer learning sessions Invite people to start early in the process Make it extremely easy to document
VALUES	

Social psychologist with HCI expertise

THE SERVICE	Diffusion of innovation • Critical mass and then others will follow

STAKEHOLDERS	• Is it coming from "us" or from "them" • Social identity theory
USE CASES	Not make too drastic changes • Small goals that are easy to achieve People are bad at estimating the effort to gain • Need to make effort look like small to make people try your thing • Effort - gain - effort - gain - etc. Boredom • Biggest reasons for majority of apps and services Theory of self regulation • Not only how I experience but how I'm telling myself of it
VALUES	• **Can't break a habit but only form a new habit to replace the old one** Context matters where you execute your habits (creates affordances) Compatibility • Is important • our system needs to work with current tools

DIY evangelist, idealist, making tools for himself

THE SERVICE	Helping others out and receiving help. Sharing and recommending interesting projects
STAKEHOLDERS	People making stuff, uninspired people wanting to make stuff, bored people
USE CASES	Knows he wastes a lot of time with useless browsing, would like to integrate looking to other peoples projects in that time to make himself more educated, useful and inspired. Imagines scrolling through some feed casually while providing meaningful feedback Want to recommend projects to people he knows, as he has done before with Makezine articles. He wants to keep updated on/part of projects where his recommendations have been integrated. Wants to share/show interest/discover other peoples projects by photographing them. He saw bike chains connecting saddles and frames in Barcelona, he wondered why, found out it was to prevent them from being stolen by asking people there. He thought it was a great little idea that needed to reach more people. He thought making pictures of

	it and uploading it to the app could help him out figure out what it was, and at the same time-share the idea. He imagines there is a place in the app where a "manifest" of goal statement can be written for projects/groups/individuals.
VALUES	Empowering

Freshly graduated arts and crafts teacher

THE SERVICE	Easy, foolproof interface, excellent feedback
STAKEHOLDERS	People with internal motivation, that can experiment together
USE CASES	Question yourself how to develop the habit of using this
VALUES	Creating and becoming part of a group of creative pioneers Safety, keeping yours what is supposed to be yours, don't let people walk away with you

E-learning specialist

THE SERVICE	"Think about the group, and how they build up their social identity. How does the media support the way they want to keep their social identity positive" Focus on what kind of observations you register, for instance the how the values in the group are created.
STAKEHOLDERS	Finding out the bias between our groups. The salience of group identity in the individual. Might be useful for explaining intergroup conflicts How do you build self-esteem for different personas?
USE CASES	Hypothesis: there is a need for a group identity and the individual identity —> people need to drive for good self esteem and others, and compare their group to other groups. (Lower the states of other groups to feel good about their own group) Social psychology theory. This might indicate that they don't want to share this with other people, or might not see it as a priority to share. "The social identity theory (SIT) Tajfel and Turner (early), Hogg and Abrams (recent) —> can be used to explain/evaluate things happening, there are survey measures, but she hasn't used them.
VALUES	What are the informal hierarchical structures

	The system supports the self-esteem on the group by giving status by the amount of shared content. (Early anthropologists) Marcel Mauss - “the Gift” What can be shared, what can be open? Creating the platform from the GROUP to feel safe, the group identity becomes stronger online, because the individual features remain unseen. De-individualization. Marc Postmes, Russell Spears, Leiah — scholar the names and compare mediated communication. “Side model of CMC”

Hackerspace core members

THE SERVICE	Possibility to use the system without a login. Integrate with other platforms, like IRC feed or Discourse. • Hacklab and its online ecosystem is very organic. • They tend not to use services that have login since it’s too big of a barrier to entry. • More than half use IRC and Etherpad (no login needed). • They started to use Discourse. • Makers jointly produce more text only content on a collection of Etherpad documents. • They have just setup a camera with eye-fi memory card that sends pictures people take with the camera to Hacklab’s Flickr account.
STAKEHOLDERS	Hacklabbers are people who don’t want to be centrally led. They want the freedom for things to happen organically. They don’t like logins. • Hacker spaces (Hacklab) and maker spaces (Kaupunkiverstas, Fab Lab) function very differently.
USE CASES	Create projects and edit them without a login.
VALUES	Online tools: Fast use, no login Make pictures to show others Collecting the loose ends left on other platforms Hacklab: Open organizational structure relying on individual’s responsible behavior Welcoming everyone, weekly opening hours Plenty of help available, and any project can be done if not obstructing others

City-owned makerspace manager

THE SERVICE	• Tried making a workflow for uploading projects, but didn't work • Light box for pictures and connection with server for optimizing data transfer • A library card system could work to track people, but she likes it that there isn't, although the bus card might be useful • Incentive? They seem they don't want to document. The KV didn't feel like making it a requirement, it would affect the experience of the space and also the content • Should be simple, straightforward. • They take pictures from themselves, but don't share it, even though that is asked • Good products are made, but not shared, which feels like a waste.
STAKEHOLDERS	o DIY culture = using traditional handiwork o Maker culture = using modern, digital technology o Maker culture is the broad spectrum, the movement is the active core which is innovating the scene o Maker movement tends to be a quite small network, the maker culture is not that big, but might come o Older women have a network to help people with tasks, there is Martat for several groups, they were really aware of the maker culture ☐ maker culture is touching existing networks, it is not perceived as a threat o Different type of users 1. Active maker, being part of the maker culture 2. Someone that needs a product o Customers are 1., but also some people having a company making some stuff for themselves (with the vinyl cutter for instance), people trying to use the 3d printer, try for free and get often hooked (making useful objects for themselves. o
USE CASES	o kv-c : introduction for 3d printing for instance, and problems, very accessible o c-c : what did you do, what did you find problematic? People are talking in the space when they are waiting ☐ there is a waiting table, but is often occupied by people with laptops working, but physical touch points are needed!, and also on the Facebook a little. Especially with the commonly used

	tools like the sewing machines there is a lot of discussion. There is a community near some tables and a group of regulars feels at home, very much in the library • Use their own designs, or use from thingiverse, not really following their steps • There is no ^license^ to ask for documentation, everything is very free
VALUES	New digital services need a reservation system, also building a community and documenting would be nice, but they cannot build the latter themselves because of resources. As example, the library is running a system that has been developed by a colleague that took initiative in his own time.

Maker event organizer

THE SERVICE	Project database • Are crawled from makezine.com, Thingiverse, etc. • Links, text, images (probably from RSS feeds or robot.txt, etc.) • Searchable and referenced Project diary • Documenting your progress • Getting help from other referenced and sibling projects • Comparing to other similar projects Interview: o Thingiverse for 3D printable objects. Linking between projects?
STAKEHOLDERS	Interview: Maybe only certain people will become contributors
USE CASES	Interview: • Physical and digital projects are different. Digital is easier to continue from previous project.
VALUES	Interview: • **Why would you want to share in the first place? What do I get?** o Collaborative projects o A tool to document the whole process as evolution of the thing instead of tasking and such o Collaborations between Hacklabs o For portfolios and personal project tracking

Science workshop instructor

THE SERVICE	The first idea of a project diary. Evolution of projects visualized and referenced.
STAKEHOLDERS	“Makers”, understood as “People that make physical things and whose projects are somehow related to online activity”
USE CASES	The tutorial can become a diary, and people can collaborate on those
VALUES	The documentation should still be accessible, flexible and transparent. But the platform is no longer having an emphasis on explaining the desired steps within the progress, but focused on the documentation of the projects of users.

Appendix B. Co-design sessions

THE FABLAB STAFF MEMBER:

Male, 25-50 y/o
Supervising activities as full-time job

Stage 1: Interview Form

Projects:

As a supervisor, guiding people with diverse projects during different stages. Revolving around proper use of tools. Responsibilities are introduction courses and "on-the-go" support.

Documentation:

Own platform for documenting projects by users:

- Basic info
- Uploaded design files
- Auto-uploading camera
- Details on materials and machine settings
    - This can be very elaborate, but it is valuable information for others and reminding yourself. This is actually asked again
- Additional info
- "step-by-step"
- Linked to Flickr and Github

Problems included:

- Spam users (added captcha)
- Technically not perfect, nobody capable of maintaining
- Paused because of problems
- Batch input at the end of projects

Other used platforms:

- Instructables: Useful, but too high bar to add content

Communication:

- Progress, Problems & Solutions
- Fabrication
- The reason WHY

Stage 2: Co-design Session

Ideas for platforms:

- Create a dynamic element for hosting our content on Fablab's static website. Our tool is embeddable.
- Finding seeds □ where does an idea come from? (Filter: Much referenced?)
- Make it fun, include an award system
- Legal policies: No weapons, illegal, dangerous, harmful or offensive materials. Document at your own risk in safe environments.

- Communicating/Exchanging WILD ideas (DO or DIE!)
- Integrating with Fablab.io for different labs.
- Inviting structure to keep up/complete your documentation (consistent quality)
- Well-designed, easy to use (usability is prime)
- Lasting networks! Events are a good thing for enthusiasm and ideas and so and therefore often happen, but results never last.
- Hosting "Good examples" like a developed python script for machine testing etc.
- Maker/Doer/Thinker
- Cycles
- Rewards for documenting Likes/Remix/I made this/References/POINTS
- Request updates from people
- Embed previews
- Make it Digestible
- Bring people up on a skill level faster
- Experts?
- Learning in early stages + Communication between experts
- Take and replicate
- Expand on ideas
- On the go markdown
- Cycle history instead of pages
- Key documents like machine specs or CNC basics
- Projects should gather interest
- Page per machine
- Long loading feeds and uncomfortable scrolling is annoying
- Materialize tacit knowledge by explaining it to others □
- Pictures are not enough

THE CRAFTS TEACHER

Male, 25 - 65
Crafts enthusiast and teaching kids

Stage 1: Interview Form

Project:

Internship project: Electronics course
Designing the program
Goal: Use laser cutter, because it can copy and students can learn the value of iteration
Existing base: Speaker making project
Troubles: Students are hard to motivate □ so iterate quickly and prototype a lot with cheaper materials. These are not teenager's primary goals. Their phones however are very primary goals.
Visited the Fablab with the class
Used Inkscape for the project
Wants to combine LEGO Mindstorm and Arduino as another project

Communication:

Background information/theory is not always used. Also teenagers want to “do” first. Theory gives a lot of added value though.
The real message is to learn exploring and learning by your self. Those are the capabilities need to survive in the world. Additionally, the understanding that “everything is made by humans”. That is what the maker community makes possible more than ever.
Many instructions were only ready in the last minute
Exhibition, but anonymous projects, because of possible bullying.
Active forums
Facebook groups

Documentation:

Intranet folder (difficult process, not usual for classes ☐ need for external web-based platform)
Student portfolios ☐ 5-10 minutes about activity at end of the class, but that didn’t work at all!
Instruction PDF’s (school doesn’t mind sharing because they don’t feel like owning it, so they basically do not care much)
Teacher has no time to document because he is giving all the help continuously

Between experts:

- Most experts stay in their preferred field
- Dedicated shop
- Facebook groups
- Forum
- Magazine
- File Area for sharing ideas
- Tradition
- Monthly meetings/seminars
- union
-

Students:

- On the phone
- Not really interested
- Feel scared of getting harsh feedback/social rejection

Stage 2: Co-design Session

- Within schools and across schools
- Attention for the “before” stage (book stand)
- The idea of the process
- Tacit knowledge
- Unified communication aids (!)
- Can progress be presented? Cherish cluelessness! Rough sketches > technical drawings ☐ create basic language
- A kid should be able to do it!
- Easy
- Stepped progress
- Too high quality will decrease the motivation
- Awarding sharing
- “it should motivate”

- Thingiverse does not support learning! Downloading-->importing-->printing is the easy way out. The goal is learning to get from a problem to a solution.
- Feed per person
- Anonymity
- Tacit knowledge should become currency

THE MAKERSPACE ARTIST

"Everyone or no-one"

Stage 1: Interview Form 1

Project: Freakstick

Super powerful magnet on a walking cane
Worked on for 7 months, now a "background project"
(With some support (read: investments) this project could go from the "freezer" to the "microwave!")
Privately funded

Communication:

Got a lot of feedback from YLE's entrepreneurial support (the loft) by product testing
Money vs. feedback ☐ both is valuable and needed

Documentation:

Could help you to identify as the "father" of this project that stimulated an active crowd
Open source: Should show people how to do it ☐ Build his visibility and make him stand out, it also brings the product further.

Environment:

Feel at home, a space to play
You cannot create if there is no nice environment (should be there 24/7)

Stage 1: Interview Form 2

Project: A lot of skulls

Preparing to become a fine artist using his experience as a 3d modeler
Likes recycling stuff

Communication:

- Kaupunkiverstas is a place where he can run into people and that can start new things happening. He can help or they can help him. (Mostly because he is talkative)

- His work is not a secret (just nobody knows) he is very open and likes sharing, he is not dependent of it, plus he is better in it, so open source is very possible.

Documentation:

- In the head ☐ Good memory! (But drinking a lot of whiskey)
- Creating an environment for making the creative space ☐ putting things on the wall and in his personal space to stay in the mood and state of mind.
- Used 3dtotal, thingiverse, tutorials on Lynda, anarchist and terrorist cookbooks

Stage 2: Co-design Session

Materials
Tools
People
Nice space
Mood boards
Storage ----------> Artist ----------> Objects!
V
V
V
V
V
Trash map and Physical space // basically an available resources inventory

Funding, Support, Feedback ----------> Artist ----------> Open Source, Teach, Sell Art
Artist --------- needs to ---------> Stand out!
V
V
V
V
More prestige = More access
Master + Pupils ☐ Responsibility increase!
Pupil > Follower

- they donate
- Spread the word
- get access to artist knowledge
- sees the “dark side”
- Private projects are still closed

Pupils as soldiers, carrying badges and are like an army. For the artist’s personal branding.

THE HACKERSPACE CORE MEMBER

Male

Stage 1: Interview Form 1

Project: XOX-Box

Analogue Synth “XOX-box”. A well-known kit in synth-builder area
About 2 days work, sourcing > building

Documentation:

Used a lot of detective work for component hunting
Vast majority of people around him don’t document or communicate anything
While working on his own project he started taking pictures of other people’s projects on a weekly basis, now there is a camera with eye-fi floating around the space and the pictures are uploaded on the blog.

- Mostly for external people to get ideas
- And telling people about association activities
- No log in, point and shoot, straight to the blog

Now there is also Discourse in use. Would not document another synth project, because he would probably solder it in one stand. However, he would report to the expert forum about the use of some special parts.

Communication:

On the special platform, about parts and the kit

Stage 1: Interview Form 2

Project: The Hacklab community
Picking up the pieces left by the board
Organizing the materials after being inspired by them
Help develop the club, which is so nice!

Communication:

IRC is around the clock busy with people a sort of collective memory. Feels like with people, not like “Publishing” on Facebook-->detached with person. It forgets (knowledge drain)
Very free organization. Do whatever you want within flexible limits
IRC Etherpad Discourse Weblog

Stage 2: Co-design Session

Personal endeavors are the most important thing to develop and create time for. Any other activity is supporting that.
The IRC Etherpad Discourse Weblog was developed organically. We should learn from that, it is very robust even though it is not perfect.

THE MEDIA STUDENT

Knows many different skills, feels the need to make stuff

Stage 1: Interview Form 1

Project: Porcelain vertebrae/Midi-controller cube

Documentation:

(She never Googled “porcelain vertebrae or bones”)
For reminding
For explaining
Instructables need to be very complete for her. Gaps cause misunderstanding.
For proof
1 Sketches of vertebrae on big paper, based on reference pictures - LOST
2 Clay model process
3 Mold process
Took pictures with phone out of pride
Facebook for group
Course blog that nobody read
Portfolio site and PDF
Presentation slides

Communication:

Only when she feels adequate, but having a lot of experience doesn’t push it either
Knowledge is in head so explanation is hard, but video helps
Teacher is support and guidance
Course blog
Facebook group
Personal diaries
Lectures
Not thinking about sharing personal projects at all, even though she understands the need

Stage 1: Interview Form 2

Project: Startup

As (product) designer
Have stages of much activity and lesser activity

Documentation:

Twitter/Facebook due to an event □ found a nice community on twitter
Trello - for designs and business stuff
Github for code
Google drive for interviews
E-mail
Text message
Skype
Website
Usually platforms are used because they are known from before, new platforms need learning
Each platform has own benefits

Communication:

Too much asynchronous collaboration is annoying
Project manager divides “cards” with tasks per member, no communication between
Needs more feedback from designer community
Urgency high □ call, text, mail, trello □ low

Stage 2: Co-design Session

Themes

- Train of thought
- Lurking for curiosity
- Team, what are the motivations and skills
- Getting expert feedback on results

Ghost mode
You choose when you reveal yourself
Talk to the ghosts
Leave a cherry bomb as a negative feedback
Transparency, show their presence

Getting expert feedback

- Expert users
- Experts in field (3d, code, design)

Goals: showing and having them test.
Messages and visits
“Packages” including the entire project files so it can be shared
Annotate content previews

Appendix C. Prototype design

Concept

This prototype is an early version of a mobile application that will be part of a multi-platform digital service. The multi-platform service will eventually consist of a mobile application and a web service.

The conceptual background on which the prototype is based is an earlier iteration primarily based on the field research, before the literature/desk research had been finalized.

This prototype was intended to allow users to:

- Document easily
- Collaborate
- Share their projects
- Search and discover
- Enhance their digital ecosystem

The functionality of the application connected to these goals as follows:

Document easily

Implemented

- **Mobile application:** The prototype was developed as a mobile application on the assumption that mobile devices are readily available to users when they need to document their activity.
- **Project logs**: List of entries that tell the story of your activity
- **Entries:** Pictures depicting or text describing your activity
- **Entry Menu**: A menu below each entry that allows you and others to comment on, reference, and see the edit history of an entry
- **Entry creation toolbar**: Minimizes the steps needed to add a new entry.
- **Various media formats**: Pictures can be taken with a camera or uploaded from phone. Text can be typed.
- **References**: Build connections to other projects.
- **Progress steps:** The steps "Plan", "Activity", and "Summary" help you create a lightweight organization within your documentation.

Planned

- Reminding notifications to keep on documenting when working on a project
- Audio recording for quick and easy text entry
- Multiple pictures in one entry
- An overview or a summary of a project
- Only show steps and maybe some kind of collage for pictures
- Moving images (GIFs)

Collaborate

Implemented

- **Commenting on entries**: You can discuss about a specific issue in the right context.
- **Project log menu**: Before entering a project you can check its details and share it with or invite others to participate in it.

Planned

- Push Notifications for new updates in projects
- Collaborative editing
- Shows who is editing and where exactly in a project
- Locks an entry in real-time when someone is editing it
- Pull to refresh project list

Share their projects
Implemented
- **Profile:** Create a profile by creating a username and connecting with your email address. Your profile is connected to your projects
- **An application link to a project** (URL): To share projects easily on various platforms

Planned
- Following projects and people
- Push Notifications
- Project view counter
- Private projects
- A link to an entry that can be pasted to e.g. a forum
- E.g. when asking for help to show the context of your question

Search and discover
Implemented
- **Search button:** visible at all times, searches the database for other projects
- **Search view:** a clean view showing your search results
- **Featured projects:** Projects that are recommended for you
- **Project log menu:** Helps you to "snowball" search from one project to another through its references

Planned
- In-app browser
- To open links from a project view
- Add references to a project directly from the browser
- Different sections of projects on the front pages
- Featured, followed, help wanted, active, etc.
- Search filtering
- Case insensitive search
- Search within open projects

Enhance their digital ecosystem
Implemented
Mobile integration: Allowing the materials being documented on a phone to be stored in a project specific place
Planned
- Browser version
- A browser URL to a project.

General improvements
Planned
- Loading animations for opening projects and other actions
- A tutorial for first time users

UI elements

Splashscreen

HOME VIEW

Navigation Header
- Search button
- Pelori Icon

Project List
- Featured projects
- Personal project logs “My Logs”
    - Project title
    - Project menu (mock-up)
        - Last edits
        - Collected References
        - Latest comments
        - Steps list
        - Settings
            - Project visibility
            - Invite new member
            - Link for sharing

Profile
- Profile picture
- Cover picture
- User name
- E-mail address

PROJECT VIEW

Navigation Header
- Back button
- Search button
- Project title

Entry Creation Toolbar
- Pictures from camera
- Pictures from phone
- Text entry
- Audio (mock-up)
- Project steps
    - Plan
    - Activity
    - Summary

Entries
- Content
    - Text
    - Image
    - Reference
    - Step
- Timestamp
- Author name
- Entry Menu
- Edits (mock-up)

        - Notification badge
- Commenting
    - Author picture
    - Author name
    - Timestamp
    - Notification badge
- Referencing
    - Reference to a project
    - Referenced by
    - Notification badge

SEARCH PAGE

Search bar

List projects of search results

- Project menu (mock-up)
    - Last edits
    - Collected References
    - Latest comments
    - Steps list

Appendix D. Prototype evaluation

Maker Faire UK

Maker Faire UK 2015 was a two day convention that took place on the 25th and 26th of April 2015 at the Life Science Centre in Newcastle UK. The convention was open to the public Saturday 25 April, 10am-6pm and Sunday 26 April, 10am-5pm.

"It's a showcase of invention, creativity and resourcefulness, and a celebration of the Maker movement. It's a place where people show what they are making, and share what they are learning. Makers range from tech enthusiasts to crafters to homesteaders to scientists to garage tinkerers. They are of all ages and backgrounds. The aim of Maker Faire is to entertain, inform, connect and grow this community." [http://www.makerfaireuk.com/about/]

Koert and Taro at Maker Faire UK at Life - A Center for world-class science.

Motivation

Our primary goal to participate on the Maker Faire was to verify the findings of our previous research by evaluating a prototype with the visitors. Additionally, by doing this we could also get in touch with potential early users for testing the beta-version later on.

The application to present our work as makers was granted at the beginning of 2015. Timo Nyberg and The Social Production research project at Aalto University have funded the trip.

Participants and visitors

With 10,000 visitors (2014) and in 120 maker stalls (2015) at Maker Faire UK there was a great variety of different kinds or makers and visitors. Based on the two days we spent talking with the visitors and makers that we have roughly categorized into groups:

Kids and Young Teenagers
*"What is this?" *Grabbing**
Accompanied by a group of peers or their parents and siblings, these curious investigators fuelled their imagination by trying out everything on the stalls and workshops, after finishing their homework.

Hobbyists
"I don't call myself a maker, but I try to make things"
One of the most common groups of visitors were the hobbyists who many times were hesitant to call them self makers as if being a maker was something you can call yourself only after achieving something more.

Enthusiast Makers
"I really like tinkering on projects, but my family is taking a lot of time at the moment, also, my DIY isn't really Maker Faire material anyway"
Many visitors seemed very knowledgeable and with a background in making things. You could see them figuring out what a stand was about pretty quickly by looking, only choosing to approach those that really grabbed their attention and fitted their background.

Professional Makers
"I'm just popping by to see what people have been working on, I usually get into more technical discussions pretty quickly if something catches my eye"
Feeling completely at ease, these makers strolled around like this was their workplace, casually talking with practically every stand-owner, as they were their colleagues, going from small talks to intense technical detail.

Educators
"We want our students to use this!"
Crafts teachers, University professors, they all came to evaluate the educational capacities of the presented technology. With only a couple more weeks until summer vacation, the Maker Faire was right on time to deliver some nice inspiration for the autumns study programs.

Corporate
"Always on the look out for innovations!"
The community is understood as a "layer of ferment" from which new enterprises can spring forth. It is important for corporations (big and small) to see where the fertile area for growth is.

Family-members of makers
"It is very nice to see all those gadgets and fun things around here!"

With their child as nr 1 priority, the maker faire presented itself as a friendly and inspiring place for many parents with kids destined to become engineers.

Categories of makers

Enthusiast Makers
"We have done a lot of experimental projects, and show them here! By the way, there is lot's of fun stuff we made and brought for you guys right over there, our contribution to the community!"
Making is not just an activity, for some it is a way of life. Through modifying, adjusting, (re-)creating or (re-)designing materials and technology around them and displaying and sharing these activities, some makers find life's purpose.

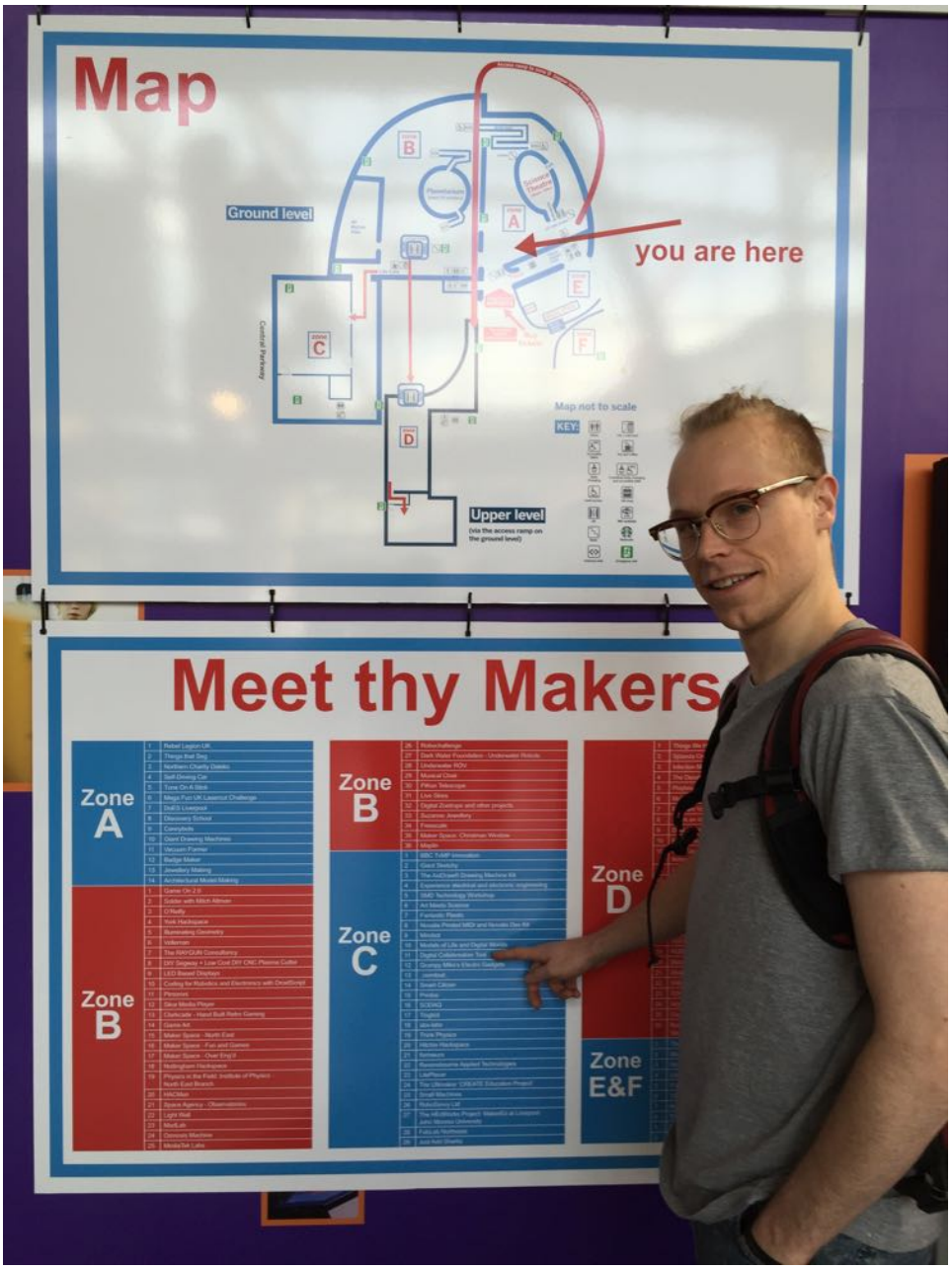
Professional Makers
"We have developed laser-cutters and are on the Maker Faire to sell them"
The maker community engages in many projects and personalities, from which some get the opportunity to grow beyond the hobbyist activity. The maker faires ensure these experts and enterprises stay in touch with their heritage and inspire new generations.

Educators
"Do you want to see how diapers can stop forest fires?" -- University stand
There were actually quite a few universities from outside of the direct area present at the convention. It was wonderful to see how the staff had translated their specialist knowledge into exciting "real-world" scenarios and workshops.

Other Creatives
*"With this satellite we can connect to the other galaxy!" *starts up artistic installation with plenty of LED's**
From performance artists to film makers, the Maker Faire attracted many creatives you would not directly expect to find at a "techy" convention. It shows that the activities involved in being a maker are more than just developing technology for pleasure or entrepreneurial practices, but can spurt forth from more artistic motives and include interdisciplinary collaboration.

Corporate
"The BBC presents some of their R&D projects on the big table in the corner"
Standing out amongst the "rubble", but in no way overly present were the big companies. Corporations such as Maplin were presenting their innovative products while blending in with the rest of the Faire.

Our booth and set-up



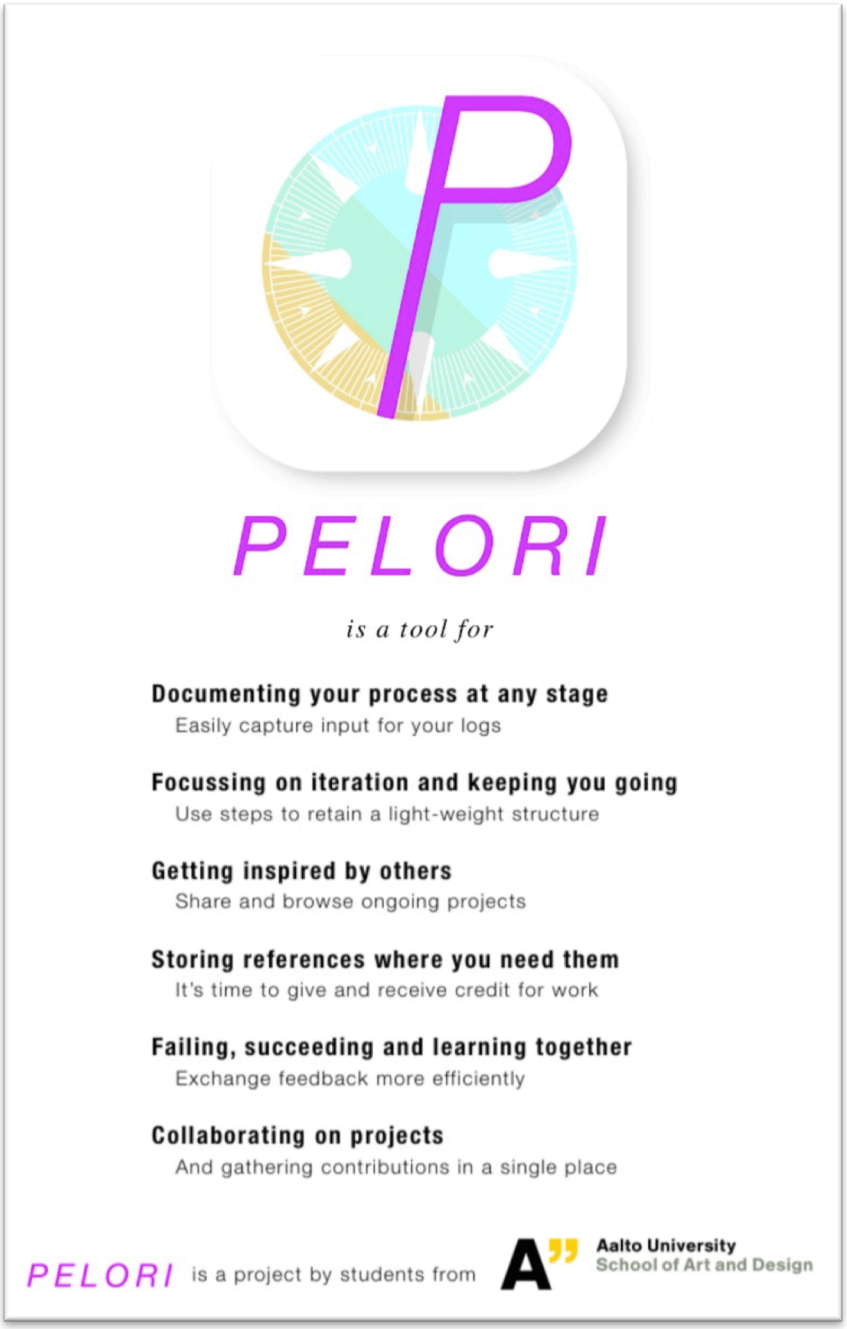
Map of the convention and our place on the itinerary

We setup our booth on Friday afternoon and dismantled it on Sunday evening. Our booth was located in the C-section of the convention space. Our space consisted of a table and a movable wall. On the table was posters and three smartphones with Pelori pre-installed. On the wall we initially had only posters. Later we made a short video to explain how the app is used and projected it on the wall.

The posters briefly explained the purpose of Pelori and invited visitors to try the app and then eventually talk with us.

Addition to our Facebook page (https://www.facebook.com/peloriapp), during the convention we also made a twitter account (https://twitter.com/PeloriApp) and published some promotional material.

Our initial booth set-up on Saturday morning



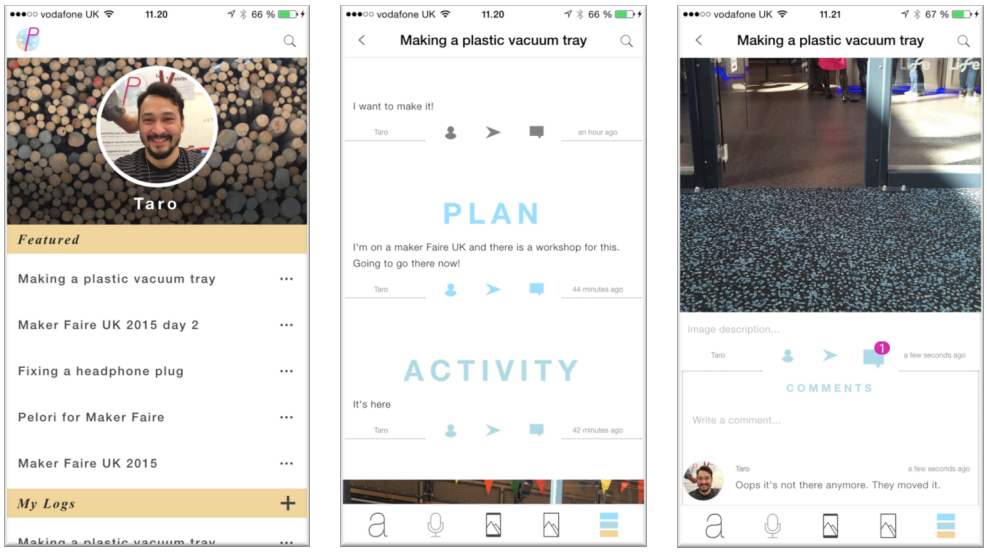
The main poster in A0 size

The Prototype

The visitors could try out the prototype on 2-3 year old flagship iOS and Android smartphones (iPhone 5, Samsung Galaxy III, Nexus 5). The devices were always on and the Pelori app was opened and ready to be used. Visitors were free to try them out and we encouraged them to pick the phones up.

The prototypes were mainly used to present our ideas and findings and as a guide for the ensuing conversation. The testing by the visitors wasn't in any way related to real world use, but we got initial reactions that were very informative. (see 'Feedback') The prototype was a good step-up to talk with makers about their documentation systems and practices.

The source code for the prototype version of Pelori that we showed at the convention can be found at:
https://github.com/taromorimoto/pelori/tree/MakerFaireUK.



The Pelori prototype at Maker Faire UK

Promotional video

On Sunday morning we decided to make a video on how Pelori is commonly used. We chose to capture the process of documenting how to make a plastic tray with embedded letters through the process of vacuum forming. There was a workshop on the Faire for doing this that gave us the unique opportunity to walk through various steps of a creative process. The raw video was edited and continuously projected on the wall with a mini LED projector. The presented video can be found at https://www.youtube.com/watch?v=ThHZ1K5YEOo.

We moved the big poster on the table, since that was where the visitors were looking at, and decided to project a looping video of Pelori on the empty backdrop.

Evaluations, feedback, beta emails

Pitching and probing

Our evaluations are based on talking to visitors and makers at our booth. These conversations usually lasted between 1 and 20 minutes. We would ask the visitors to explain how they documented their work and how it usually was a struggle. We then proceeded to explain how we identified the problem and we thought Pelori could solve this using the prototype. During talking they got to try out Pelori on the test devices and we also showed them different ways the app could be used. With the conversation focused strongly on the intersection of their documentation practice and the abilities of the app, the feedback was useful for improving our formulation of the problem as well as the intended solution.

Koert talking to visitors who are testing the prototype

Gathering core users

On our table we had a notebook and a pen where the visitors could write their email addresses if they were interested in more information and testing later versions of the application. In total 43 people gave their email address to sign up for more information and participate in beta testing. Since we have had the (sometimes quite extensive) introduction and discussions about the problem and the app before we would ask people to enlist we believe that we will get a relatively high yield of these addresses.

Findings

We decided to document the observations, citations and abstracts of the discussions with written down notes during or directly after the conversation. We have abstracted the findings and listed them below.

Observations

“I saw makers make a group picture with one of their projects, they might want to tag themselves, as well as their projects in that picture”.
It helps to relate this app to other apps when explaining.
I feel the need to have a project printed out to show them.
At this point, nobody is really getting the steps.
Notifications can be thought of much better.
People refer to this as a timeline.

Step menu is a bit light in the environmental light.
Going back and forth between phone and desktop is regarded as really good.

Feedback

Age	Type	Feedback
Older man	Not a typical maker	Reminds me of Prince project management methodology
Elderly woman	Ceramic artist	Picture annotation, printing a picture and using lines to point out several areas on an image and text to comment on them
		“Glaze will completely destroy your phone! A friend of mine made hers totally unusable”
Teenager	Young maker	Safety measures are obstructive etc. on the way of documentation
		Dictation or audio could help
Middle age man	Journalist	Benchmark tips, “reporting” apps, very image heavy:
		Storehouse
		Together
		Steller
		He thought “Medium” was too text-heavy
		Check out @journalism21 for more reviews on journalism tools
Young boy		“Are steps for multi-tasking?”
		After extensive scrolling and reading “I would use this for leisure time”
Man	Corporate employee	“Could an enterprise solution for corporate teams be installed on a company’s own servers?”
25 year old man	Maker	“What is the difference between this and Facebook groups?”
		“I read a lot on Reddit”
35 year old father	Maker	“Planning is very difficult and I have no time for documenting because I have 2 kids of 2 and 3 year old”
3 middle age men	High school teachers	“We want our students to use this!”
		“They cannot keep a blog even if we ask them to”
		“They all have smart phones”
Middle	Chemistry	“I usually annotate pictures to show my students what

age man	teacher	I mean, but I sometimes need to print them to be able to do that"
		Text on paper to digital text conversion?
Man in his 30s	Filmmaker	G-cal notes to calendar entries
		"Export to pdf?" (Different export formats)
		Notifications/suggestions when in an area. Relevance is determined by place, and people/projects etc. are recommended
		Hipchat : lots of group projects + general chats with team members
		Overview of progress -> tracking "what" "who" "where" "when" all for ideation -> retrieve steps
		Parallel - sub goals, closing and continuing on those
	Various many people	"So I can follow stuff from others?"
	?	How can we connect the "non-smart" crowd (desktop?)
45 year old man	Professional maker	"I would use it"
	?	"This would be good for schools!"
Man in his 40s	Professional maker	"I had a project in which I designed and made a USB controlled pedal for a computer. After a couple of months someone wanted to buy one from me, but I had already completely forgotten how to make it, and I hadn't documented how I had done it. SO I had to redo the entire thing from scratch. The only thing I had was a picture of the wiring. That was a very important picture."
30 year old couple	Enthusiast maker couple	Man: "Every maker has a billion projects, but only 1% gets finished, a lot actually fails"
		"I make pictures of what I do, but they are all in my camera roll, with thousands of other pictures"
		His girlfriend, also maker, but not as experienced:
		"For me it is really good to hear and know that I don't have to successfully complete everything"
50 year old man	Hobbyist	"Dictation with pictures would be cool"
		"I want something that is continuously documenting, and if I did something I want to keep, I can ask, keep the last 2 minutes, and it will only save that"
		"I also want to be able to see back what I did in those two minutes in slow motion, for if I didn't understand

		what I did”
10-14 year old boys	Bunch of school kids	Got the app pretty fast, 2 of them had the exact same phones as our test devices, so they could basically run this in school too:
		“I would pay 99p for this in the app store”
	?	Storage could be done on Google drive, but it could be referenced inside the app.
	?	Picture taken □ straight upload? It might be nice to take it more quickly, but if it is un-sharp, it will be bad.
	?	“How do you say that you are done with a project?”
	?	“Is it like a collaborative Workboard?” (On inspection, Workboard is a management app”
45 year old father	Hobbyist maker	“This would be really good for mechanical bits, when I need to take apart and reassemble a complex structure with lots of bits and pieces, this could help me to keep track of what should be where, so I can see where that tiny thing that is left in the end should have actually been”
		“It could also be really good for keeping track of the recipe you are making, just photographing and annotating all the ingredients and the steps that you are doing”
Man in his 40s	Maker	Uses a self made note-taking app, but actually uses it mostly to record his working times.
Father in his 40s	Professional Maker (python + products)	“I need to record what was working at what stage, and relate the physical things to the code, so I can understand the 1500 lines of code that I wrote and see what was intended to work where and why”
		“I want to use this, this would save me a lot of time”
Teenage boy	Test with young enthusiastic maker (See Cyber Eye)	Intro was too much typing
		Typing was a no go at all
		He uses speech to make essays for school
		Glasses and safety measures are in the way
		Descriptions could be done later (typing was annoying, this is a quick fix, would he actually do it later?)
		Search was too slow to be effective
		He got really frustrated and angry when his projects were failing, how does our app relate to those emotions?
Man	Maker	“If my spelling is bad, I might want to keep my

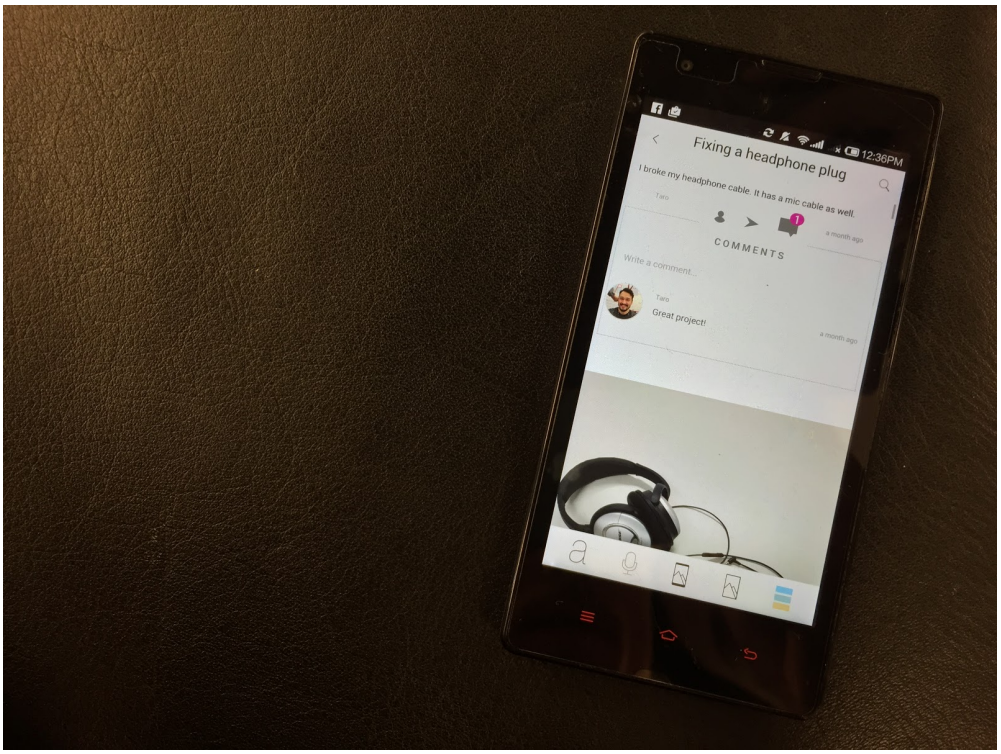
		projects private”
30 year old man	Professional maker, art science background	Free Dropbox is usually enough, functionality integrates greatly with Google groups. This is often used in the hack community (webiste-dropbox-googlegroup)
		“I see this could be an Instagram for hackers!”
		“it would be a great place to collect all the half-finished projects in one space”
		“art science needs to get funding, so we need to communicate our actions with non-makers with money”
	?	Trello: Google docs : IRC

Demo Day

We also discussed the prototype with makers on the 2015 Spring Demo Day at the Media Lab. It’s a 5-hour open event for Media Lab researchers and students to exhibit their work through presentations on stage and demoing on stands. The attending 200-300 visitors were students, teachers or professionals in the field of new media.

http://medialab.aalto.fi/about/x-demodays/2015-spring-demo-day-programme/

Our setup for the stand and process for getting feedback was identical to the one at the Maker Faire. The presentation was a 3-minute summary of the project and the prototype.



Picture of pelori from the Demo Day presentation.

Feedback

Age & Gender	Type	Feedback
Male 30s	Design student	Makes music. Wants an easy play button for audio recordings.
Woman in mid 20s	Student	Want's to reorder entries.
Woman 30s	Researcher or teacher	Want's to use for Design Factory courses such as PDP.
Men in 20s-30s	Students	The Verge and Techcrunch use live blogging to let multiple journalists to submit entries that then get's edited by one editor.
Woman in mid 25s	Student	"Photo and photo library icons should be combined into one icon."
		"The app workflow could be minimized even more."
Woman 30s	Student	Should be able to re-organize photos/entries.
		New entries should appear where the project was viewed (scroll).
		Project list should have created date.
		Want's exact dates and times instead of humanized format. Can also be humanized for one day old and data/time for older.
		Private projects for companies.
		Good for company travel reports to show others.
2 woman 30s	Media professionals	"We could use this to improve our picture exchange which is usually done with messaging apps. This causes material to get lost."
		"Just finish this fast so we can use it!"
		"It would be great to monitor students and comment on their progress."
Male 25	Student or designer	Podio --> Benchmark
		Asana --> Looks ugly

Appendix E. Beta version design

Here is a list of elements that was implemented and plan to be implemented later after the beta version. Features actually implemented during the one-week implementation phase are marked with "[implemented]" text.

Elements planned to be implemented

Remove steps [implemented]

- Steps create more complexity and in so are getting in the way of documenting.

Minor fixes and changes [implemented]

- Remove image captions
- Remove project info entry
- Save photo also to the camera roll. (bug: failed to work out of the box)
- Private slider message
- Flip the order of the project entry list
- Always open the app in the last viewed project

Optimization

- Create entries first and save in the background! (no waiting after tapping save/create)

GIFs [implemented]

- Capture a GIF and add it as an entry (missing iOS and Safari support, Android buggy)

Browser version

- Browser version would be useful for adding lots of text, code and links.
- These users could be teachers, students, coders, and researchers.

Toolbar re-design [implemented]

- GIF icon
- Add project icon
- Visible and slightly different in the all views

Comments

- Comments should be refactored into sub-entries
- Comments in new window
- Comment with all content types

Sign-in [implemented]

- New page for first time users
- No passwords, only a link to your email that allows you to login (email authentication was replaced with Facebook login)

Hamburger menu [partly implemented]

- Filter options
- Remove projects and entries
-

Referencing/connecting projects [partly implemented]

- Project level referencing instead of fine grain entry based.
- Can create sub logs
- Open projects
    - On a conceptual level these are basically collections or groups.
    - Technically these are projects that are editable by all.
    - Users can follow (think group)
    - Everyone can add projects to this project (think collection)
    - Can also see log of entries in subprojects
    - Activity feed of important events
    - Various different uses: Hacklab Helsinki, CNC machines, light fixtures, etc.

Following

- Follow projects and people

Combine take a photo and get picture from gallery

Projects

- Viewing time for each project instead of view count

Profile

- GIFs as profile pictures
- Find other people's profiles

Push Notifications

- Weekly summary: People viewed your projects for x minutes.
- Get notification from a comment on entry you commented.
- When specifically interested in a followed project or a user, can turn notifications on for them.
- Send a notification to after tapping an "!" icon in an entry of a project user owns.
    - Only users who have turned project notifications on.

Show on-going projects, failing is good [implemented]

- Design to document easily
- No way to polish content or reorder it
- Show picture and short text during signup to convey the central idea that failing and unfinished is good.

Annotate on an image similarly to Facebook tagging.

- This would happen through commenting.

Planned to be implemented later

Offline use

- It should be possible to use that app without Internet connection.

Integrate to other services

- Embed a project log onto a web page
- Integrate feeds from other services into a project

Exporting

- Export a project as a PDF and send to an email address.

Private projects

- Project visibility toggle
- Only a paid option like GitHub

Push Notifications

- Interval reminder for documenting

Appendix F. Example persona

The example persona shown in this appendix is direct copy from Koert's unfinished thesis.

Persona 2: The Shepherd

Educational goals
The shepherd's goal is to introduce making to new audiences. He takes on the role of a teacher and wants to communicate lessons through projects. A project could for instance revolve around the importance of iteration of the value of theory for making. He develops these projects with great care and takes into account that his audience has little to no experience with making.

Collaborative activity
He is part of a community of shepherds that has a long history and has formed a union. The union has helped information exchange between shepherds for decennia, and has its own publication and online communication platforms. The union is divided in several informal groups based on expertise. Shepherds' projects are usually connected to their specialization, even though they feel the need of introducing many different skills to the audiences. This is why they within the union projects are actively shared across disciplines.

Design
The shepherd identifies two important design directions for audiences that are new to making, these are:

- Social pressure is bad, with this is meant that new makers are insecure about their projects and are easy to discourage. For instance, if they feel the quality of their creation is too low they will not want to share it for the fear of being bullied. The shepherd did find that the audience feels comforted when they are able to exhibit their work anonymously. The shepherd thinks a successful communication platform should have an environment in which cluelessness is cherished.

- Basic and unified communication. According to the shepherd there should be easier ways to document and share information. What he thinks is of key importance is the development of rough standards that could guide users to document quickly and thoroughly throughout their process.